Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 1740572, 16 pages
https://doi.org/10.1155/2017/1740572

Research Article

A Simple Provably Secure AKE from the LWE Problem

Limin Zhou[1] and Fengju Lv[2]

[1]*Information Security Center, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2]*The Seventh Middle School of Zibo, Shandong 255499, China*

Correspondence should be addressed to Limin Zhou; zhoulimin.s@163.com

Received 20 November 2016; Accepted 12 March 2017; Published 19 April 2017

Academic Editor: Bruno G. M. Robert

We first propose an authenticated key exchange (AKE) from the LWE problem. The AKE is simple since it does not involve any other cryptographic primitives to achieve authentication and depends on solely the LWE problem in the worst-case (e.g., SVP and SIVP). We give its security under the BR model, which captures wPFS and might be appealing in specific applications.

1. Introduction

Authenticated key exchange (AKE) is one cryptographic tool in establishing secure communication channels which provide secrecy and authenticity for both communication parties. AKE not only allows parties to utilize their known information to compute a session key which is unknown to anybody except for parties, but also ensures authenticity of communication parties, so that an adversary cannot impersonate one party in the conversation.

In an AKE, each party has a *static public key* which is produced by its *static secret key* and certified with a party's identity through a public key. Communication parties utilize *ephemeral secret keys* to generate *ephemeral public keys* and compute a *session state* from their *static public/secret keys*, the *ephemeral public/secret keys*, and the transcripts of the session. Communication parties then obtain a *session key* from these values using a *key derivation function*.

The *session key* guarantees data integrity and confidentiality, which implies that security notion for AKE should be developed. To handle this case, Bellare and Rogaway [1] first provided BR security model for AKE which was based on indistinguishability. The BR95 [2] and BPR2000 [3] models were extensions to the BR93 model. Although the BR model captured key authentication, for example, *confidentiality* of session keys, and basic security requirement, for example, known key security and impersonation resilience, it cannot grasp more complex scenes if one party's *static secret key* or *session state* was revealed. Accordingly, Canetti and Krawczyk [4] defined the first Canetti-Krawczyk CK model which grasped the leakage of static secret keys and session state. But it was not resilient to advanced attacks, for example, key compromise impersonation (KCI) and perfect forward secrecy (PFS) which guaranteed an adversary not obtaining the session key after a completed session even if the static private keys of the parties were subsequently revealed. To resist advanced attacks, Krawczyk [5] proposed HMQV protocol in the CK+ model (which was stronger than CK model [6]) and showed that no 2-pass AKE achieved PFS. Alternatively, he presented weak perfect forward secrecy (wPFS) which guaranteed security only for previous sessions without an adversary's intrusion. Namely, wPFS declared that the session key was still private if the static keys of a completed session were revealed [5]. To modify the CK+ model, LaMacchia et al. [7] and Sarr et al. [8] proposed the eCK model (which was not stronger than the CK model) and the seCK model, respectively. This paper will only show AKE security under the BR model [1].

In the past three decades, there appeared a large number of AKEs based on number-theoretical problems [9, 10]. With the rapid development of computing technology, for example, quantum information technology, quantum computer brought great threat to these protocols based on classic number-theoretical problems. With a quantum computer, quantum polynomial time algorithm [11] for factorization and the discrete logarithm problem had brought challenges

for these traditional cryptosystems. Recently, researchers plan to focus on quantum resistant cryptographic primitives. Lattice-based cryptosystem was one potential candidate for postquantum.

To date, there existed a lot of lattice-based cryptosystems [12–16] because lattice-based cryptosystems can capture strong security proof based on worst-case hardness assumption that can resist quantum attack and be implemented efficiently. What is more, most of lattice-based cryptographic constructions [12, 15, 16] were based directly upon one of the two average-case problems that had been shown to enjoy worst-case hardness guarantees: the Small Integer Solution (SIS) problem [12, 15] and the (Ring-) Learning with Errors problem [13, 14, 16].

As mentioned above, in view of the security guarantee against quantum adversaries, there had been a great number of lattice-based cryptosystems [12–19], which offered resilience against quantum computer attack. Cryptographers especially had put effort into constructing various key exchanges (KEs) and AKEs from the (Ring-) LWE problem, for example, lattice-based KE [20] which can only be secure in passive model but made a big step in constructing a post-quantum KE and a NTRU-KE based on Ring-LWE [21], as well as lattice-KEs [22–25]. However, we only know a few of results on lattice-based AKEs [17–19, 24, 26, 27]. What is more, Ding et al. presented an attack with the leakage of the signal function [28] on RLWE based KE [20]. Gong and Zhao presented a small field attack (SFA) [29] on the one-pass protocol [24]. Motivated by post-quantum security, our paper will focus on the construction of a lattice-based AKE based on the LWE problem [13, 14]. Our basic AKE is simple and comes with a rigorous proof of security based on the LWE problem under the BR model. The AKE is simple since it does not involve any other cryptographic primitive to achieve authentication and depends solely on some hard lattice problems in the worst-case (e.g., SVP and SIVP). We prove its AKE security with wPFS under the BR model.

Ding et al. [28] showed that KE based on RLWE problem could be broken by analyzing the number of signal changes of each of the coefficients. Ding and Lin [20] utilized the signal function to construct a KE from (Ring-) LWE. Theoretically, the KE from LWE [20] could be broken by the attack with the leakage of the signal function [28], as the KE from (Ring-) LWE [20] only referred to matrix-vector multiplication in finite field. Hence, our proposed lattice-based AKE from the LWE problem could suffer from the same attack with leakage of signal function [28]; here we do not study it and omit it. Gong and Zhao exploited a SFA (with a property of the CRT basis of R_q, i.e., Proposition 5 in [29]) against one-pass AKE [24] although the SFA may not violate the security of one-pass AKE [24]. Notice that SFA [29] applied only to a special case of the original Ring-LWE problem [16] which sufficed for [24]. Likewise, maybe there exists a similar SFA (with the help of some properties) to break our proposed AKE since every cryptosystem will be broken in the future. And we do not know whether SFA can be applied to the LWE problem since Ring-LWE problem is one special case of the LWE problem [13, 14]. For example, cyclotomic polynomial [30] which was essential for SFA [29] applied only to polynomial ring. We leave them as open problems. Maybe our AKE could capture AKE security and resist some advanced attacks under the CK model, the CK+ model, or the eCK model, but we leave them as future works.

This paper is organized as follows. Section 2 contains definitions and properties related to lattice. In Section 3, we construct a lattice-based AKE based on the LWE problem. Section 4 gives its AKE security under the BR model. Section 5 gives comparison. Conclusion is in Section 6. The BR model is given in appendix.

2. Preliminaries

Notations. Assume that n is the main security parameter. Let notations be as defined in [13, 14]. Let Λ be a discrete subset of Z^m. The *Gaussian function* on $\mathbb{R}^m$ centered at $c \in \mathbb{R}^m$ with any positive $\sigma \in \mathbb{R}$ is $\rho_{\sigma,c}(x) = \exp(-\pi(\|x - \mathbf{c}\|^2/\sigma^2))$, $\forall x \in \mathbb{R}^m$. Let $\rho_{\sigma,c}(\Lambda) = \sum_{x\in\Lambda} \rho_{\sigma,c}(x)$ be the discrete integral of $\rho_{\sigma,c}$ over Λ and $\mathcal{D}_{\Lambda,\sigma,c}$ be the *discrete Gaussian distribution* over Λ with center c and parameter σ. Concretely, $\forall c \in \mathbb{R}^m$, $\forall \sigma \in \mathbb{R}$, define $\mathcal{D}_{\Lambda,\sigma,c}(\mathbf{x}) = \rho_{\sigma,c}(\mathbf{x})/\rho_{\sigma,c}(\Lambda)$, $\forall x \in \Lambda$. If $c = 0$, $\rho_{\sigma,0}$ and $\mathcal{D}_{\Lambda,\sigma,0}$ are shorted for ρ_σ and $\mathcal{D}_{\Lambda,\sigma}$, respectively.

Regev proposed the *Learning with Errors (LWE) problem* [13].

For integers $n \geq 1$ and $q \geq 2$, $\alpha \in (0, 1)$. Let $\mathbf{s} \in \mathbb{Z}_q^n$, $A_{s,\alpha}$ be the distribution on $\mathbb{Z}_q^n \times \mathbb{Z}$ obtained by choosing a vector $\mathbf{a} \in \mathbb{Z}_q^n$ uniformly at random and a noise term $e \leftarrow D_{Z,\alpha q}$, and output $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s}\rangle + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$.

The LWE problem is as follows: for uniformly random $\mathbf{s} \leftarrow Z_q^n$, given poly($n$) number of samples that are either from $A_{s,\alpha}$ or uniformly random in $Z_q^n \times Z_q$, output 0 if the former holds and 1 if the latter holds.

The decision LWE problem is at least hard as approximating several problems on n-dimensional lattice in the worst-case within $\widetilde{O}(n/\alpha)$ factors using a quantum computer [13] if $\alpha q \geq 2\sqrt{n}$, $q = \text{poly}(n)$. Brakerski et al. [31, 32] showed that the LWE assumption still preserved if $b = \langle \mathbf{a}, \mathbf{s}\rangle + te$ for $\forall t \in \mathbb{Z}^+$ and $gcd(q, t) = 1$ but security loses with a $\sqrt{n}$ factor. The *HNF-LWE assumption* [32] declared that HNF-LWE problem was still hard if the secret came from the error distribution; for example, $s \leftarrow D_{z^n,\alpha q}$.

Formally, a random noise vector with a Gaussian distribution is used to prove that certain lattice problems are in coNP [33]. Lemma 1 [33] gives a norm bound of Gaussian distribution.

Lemma 1 (see [33]). *For any n-dimensional lattice Λ, a vector $c \in R^n$, and reals $0 < \epsilon < 1$, $s \geq \eta_\epsilon(\Lambda)$, we have*

$$\Pr_{x\sim\mathcal{D}_{\Lambda,s,c}}\{\|x - c\| > s\sqrt{n}\} \leq \frac{1+\epsilon}{1+\epsilon} \cdot 2^{-n}. \tag{1}$$

Signal Functions [20]. Define the signal function discussed in [20]. For prime $q > 2$, given $Z_q = \{-(q-1)/2, \ldots, (q-1)/2\}$, $E = \{-\lfloor q/4\rfloor, \lfloor q/4\rfloor\}$, define σ as the signal function in E: $\sigma(v) = 0$ if $v \in E$ and 1 otherwise as follows.

For simplicity and requirements in some places, set

$$\mathrm{cha}(v) = \sigma(v) = \begin{cases} 0, & v \in \left[-\left\lfloor \frac{q}{4} \right\rfloor + 1, \left\lfloor \frac{q}{4} \right\rfloor\right] \\ 1, & \text{otherwise.} \end{cases} \quad (2)$$

For any $v \in Z_q$, $v + \mathrm{cha}(v) \cdot ((q-1)/2) \bmod q$ belongs to $[-\lfloor q/4 \rfloor + 1, \lfloor q/4 \rfloor]$.

We define *modular function* $\mathrm{mod}_2(v, \omega)$ from $Z_q \times \{0,1\}$:

$$\mathrm{mod}_2(v, \omega) = \left(v + \omega \frac{q-1}{2} \bmod q\right) \bmod 2, \quad (3)$$

where $v \in Z_q, \omega \in \{0,1\}$.

Modular function was discussed as *robust extractor* [20] and can guarantee the correctness of our protocol.

Lemma 2 (see [20]). *Let $q > 8$ be an odd integer; the function mod_2 defined above is a robust extractor with respect to ω with error tolerance $q/4 - 2$.*

Lemma 3 (see [20]). *For any odd $q > 2$, if v is uniformly random in Z_q, then $\mathrm{mod}_2(v, \omega)$ is uniformly random conditioned on $\omega \in \{0,1\}$.*

Lemma 4 (see [24]). *Let n be the security parameter and odd prime $q = 2^{\omega(\log n)}$. For any $b \in \{0,1\}$ and $v' \in Z_q$, the output distribution of $\mathrm{mod}_2(v + v', b)$ conditioned on $\mathrm{cha}(v) \in \{0,1\}$, where the probability is taken over the uniform and independent choice of $v \in Z_q$.*

3. One AKE from the LWE Problem

Let $n \geq 1$ and $q \geq 2$ be integers, $\alpha \in 0, 1$, q is prime. For the same integer k, let $h : \{0,1\}^* \to \{0,1\}^k$ be KDF which is modeled as random oracles.

Sample a uniformly random matrix $M \leftarrow Z_q^{n \times n}$. Let $p_A = Ms_A + 2e_A \bmod q \in Z_q^n$ and s_A be the static public key and static private key of Alice ($\overline{A}$), where $s_A, e_A \leftarrow \mathcal{D}_{\mathbb{Z}^n, \alpha q}$. Let $p_B = s_B^t M + 2e_A^t \bmod q \in Z_q^n$ and s_B be the static public key and static private key of Bob ($\overline{B}$), where $s_B, e_B \leftarrow \mathcal{D}_{\mathbb{Z}^n, \alpha q}$. Assume that the protocol works between Alice and Bob.

Setup. Alice randomly chooses $r_A, e_1 \leftarrow \mathcal{D}_{\mathbb{Z}^n, \alpha q}$, computes $R_A = Mr_A + 2e_1 \bmod q$, and sends R_A to Bob.

Response. Upon receiving R_A from Alice, Bob randomly chooses $r_B, e_2 \leftarrow \mathcal{D}_{\mathbb{Z}^n, \alpha q}$, computes $R_B = r_B^t M + 2e_2^t \bmod q$, $k_B = r_B^t \cdot p_A + s_B^t \cdot R_A \bmod q$, and $\omega_B = \mathrm{cha}(k_B) \in \{0,1\}$, and sends (R_B, ω_B) to Alice. Then, Bob computes $\sigma_B = \mathrm{mod}_2(k_B, \omega_B) \in \{0,1\}$ and derives $sk_j = h(\overline{A}, \overline{B}, R_A, R_B, \omega_B, \sigma_B) \in \{0,1\}^k$.

Completion. Upon obtaining (R_B, ω_B), Alice computes $k_A = p_B \cdot r_A + R_B \cdot s_A \bmod q$, $\omega_A = \mathrm{cha}(k_A) \in \{0,1\}$, $\sigma_A = \mathrm{mod}_2(k_A, \omega_A) \in \{0,1\}$, and $sk_B = h(\overline{A}, \overline{B}, R_A, R_B, \omega_A, \sigma_A) \in \{0,1\}^k$.

Correctness. If Alice and Bob run the protocol honestly, they will share the same session key. To show the correctness of our AKE, it is sufficient to show that $\sigma_A = \sigma_B$. σ_A and σ_B are output by mod_2 with the same second input $\mathrm{cha}(k_B)$. According to Lemma 2, we only show that k_A and k_B are sufficiently close.

If $8(\alpha q)^2 \cdot n \leq q/4 - 2$, then $k_A = k_B$ with overwhelming probability.

Proof. From the form of k_A, k_B,

$$\begin{aligned} k_A &= p_B \cdot r_A + R_B \cdot s_A \\ &= \left(s_B^t M + 2e_B^t\right) \cdot r_A + \left(r_B^t M + 2e_2^t\right) \cdot s_A \\ &= s_B^t M \cdot r_A + 2e_B^t \cdot r_A + r_B^t M \cdot s_A + 2e_2^t \cdot s_A \\ &= s_B^t M \cdot r_A + r_B^t M \cdot s_A + 2e_B^t \cdot r_A + 2e_2^t \cdot s_A \\ &\quad \bmod q \\ k_B &= r_B^t \cdot p_A + s_B^t \cdot R_A \\ &= r_B^t \cdot (Ms_A + 2e_A) + s_B^t \cdot (Mr_A + 2e_1) \\ &= r_B^t \cdot Ms_A + 2r_B^t \cdot e_A + s_B^t \cdot Mr_A + 2s_B^t \cdot e_1 \\ &= r_B^t \cdot Ms_A + s_B^t \cdot Mr_A + 2r_B^t \cdot e_A + 2s_B^t \cdot e_1 \\ &\quad \bmod q, \end{aligned} \quad (4)$$

we obtain

$$\begin{aligned} k_A &= k_B + 2r_B^t \cdot e_A + 2s_B^t \cdot e_1 - \left(2e_B^t \cdot r_A + 2e_2^t \cdot s_A\right) \\ &\quad \bmod q \\ &= k_B + 2\left(r_B^t \cdot e_A + s_B^t \cdot e_1 - e_B^t \cdot r_A - e_2^t \cdot s_A\right) \\ &\quad \bmod q. \end{aligned} \quad (5)$$

By Lemma 1, we have

$$\begin{aligned} &\left\| 2\left(r_B^t \cdot e_A + s_B^t \cdot e_1 - e_B^t \cdot r_A - e_2^t \cdot s_A\right) \right\| \\ &\quad \leq 8 \cdot (\alpha q \sqrt{n}) \cdot (\alpha q \sqrt{n}) = 8(\alpha q)^2 \cdot n \leq \frac{q}{4} - 2 \end{aligned} \quad (6)$$

with overwhelming probability. That indicates k_A and k_B being sufficiently close.

By Lemma 2, with mod_2 with respect to S with error tolerance $q/4 - 2$, we have

$$\sigma_A = \mathrm{mod}_2(k_A, \omega_A) = \mathrm{mod}_2(k_B, \omega_B) = \sigma_B. \quad (7)$$

Further, we show that

$$\begin{aligned} \sigma_A = \sigma_B &= r_B^t \cdot Ms_A + s_B^t \cdot Mr_A + \frac{q-1}{2} \cdot \omega_B \\ &= \left(r_B^t \cdot Ms_A + s_B^t \cdot Mr_A + \frac{q-1}{2} \cdot \mathrm{cha}(k_B) \bmod q\right) \\ &\quad \bmod 2. \end{aligned} \quad (8)$$

This is because

$$r_B^t \cdot Ms_A + s_B^t \cdot Mr_A + \mathrm{cha}(k_B) \cdot \frac{q-1}{2} \mod q$$
$$= k_B + \mathrm{cha}(k_B) \cdot \frac{q-1}{2} - \left(2r_B^t \cdot e_A + 2s_B^t \cdot e_1\right) \mod q \quad (9)$$

and $|k_B + \mathrm{cha}(k_B) \cdot ((q-1)/2)| < q/4 + 1$.

Hence, we have

$$r_B^t \cdot Ms_A + s_B^t \cdot Mr_A + \mathrm{cha}(k_B) \cdot \frac{q-1}{2} = k_B + \mathrm{cha}(k_B) \cdot \frac{q-1}{2} \mod q - \left(2r_B^t \cdot e_A + 2s_B^t \cdot e_1\right), \quad (10)$$
$$\sigma_A = \sigma_B = \left(r_B^t \cdot Ms_A + s_B^t \cdot Mr_A + \mathrm{cha}(k_B) \cdot \frac{q-1}{2} \mod q\right) \mod 2.$$

□

Parameter Selection. Here select the same parameters as those in [20]: $n = \lambda$, $q = \lambda^4$, $\alpha = 1/\lambda$. It is easy to verify that $\alpha q \geq \sqrt{n}$ and the correctness holds.

4. Security

In our AKE, the public matrix $M \leftarrow Z_q^{n \times n}$ is public and every *static public key* actually consists of a LWE tuple with Gaussian parameter α. Thus, the *static public key* is computationally indistinguishable from a random element in Z_q^n under the LWE assumption. Analogously, R_A and R_B are also computationally indistinguishable from random elements in Z_q^n under the LWE assumption with Gaussian parameter α.

To show the randomness of the session key, it is enough to take Bob as an example. Obviously, if k_B is random over Z_q, σ_B is statistically close to $\{0, 1\}$ even conditioned on ω_B by Lemmas 2 and 3. Note that h is a random oracle; thus sk_B is uniform over $\{0, 1\}^k$. Now, we check the randomness of k_B:

$$\begin{aligned} k_B &= r_B^t \cdot p_A + s_B^t \cdot R_A \\ &= r_B^t \cdot (Ms_A + 2e_A) + s_B^t \cdot (Mr_A + 2e_1) \\ &= r_B^t \cdot Ms_A + 2r_B^t \cdot e_A + s_B^t \cdot Mr_A + 2s_B^t \cdot e_1 \\ &= r_B^t \cdot Ms_A + s_B^t \cdot Mr_A + 2r_B^t \cdot e_A + 2s_B^t \cdot e_1 \mod q. \end{aligned} \quad (11)$$

It is necessary to establish the randomness of k_B on the hardness of the (decisional) LWE problem, since p_A, R_A are actually LWE instances, and r_B^t, s_B^t are random elements in Z_q^n under the LWE assumption with Gaussian parameter α. Generally, we will prove that k_B is statistically close to a real LWE instance if the secret and the error are randomly from $\mathcal{D}_{\mathbb{Z}^n, \alpha q}$. Since $p_A = Ms_A + 2e_A \bmod q \in Z_q^n$ and $R_A = Mr_A + 2e_1 \bmod q$ are random over Z_q^n, thus

$$\begin{aligned} k_B &= r_B^t \cdot p_A + s_B^t \cdot R_A \\ &= r_B^t \cdot Ms_A + s_B^t \cdot Mr_A + 2r_B^t \cdot e_A + 2s_B^t \cdot e_1 \mod q \end{aligned} \quad (12)$$

is random over Z_q. That is, k_B is statistically close to a real LWE instance.

Formally, let N be the maximum number of parties and m be maximum number of sessions for each party. We separate the security proof for the initiator and responder, respectively.

4.1. Security for the Initiator. In this session, the AKE security for initiator is proved when the initiator is the owner of the test session. Let $sid^* = (\Pi, I, A^*, B^*, R_{A^*}, (R_{B^*}, \omega_{B^*}))$ be the test session chosen by the adversary $\mathcal{A}$. Consider two types of adversaries.

Type One: R_{B^*} is output by a session activated at B^* by a $Send_1(\Pi, R, B^*, A^*, R_{A^*})$.

Type Two: R_{B^*} is not output by any session activated at B^* by a $Send_1(\Pi, R, B^*, A^*, R_{A^*})$.

To capture wPFS, *Type One* adversary is allowed to obtain the static secret keys of parties A^* and B^* by corrupting A^* and B^* (but *Type Two* adversary is not allowed to corrupt either A^* or B^*).

4.1.1. Security against Type One Adversary. First, AKE security is proved for any PPT *Type One* adversary $\mathcal{A}$.

Theorem 5. *If HNF – $LWE_{q,n,\alpha}$ is hard, the proposed AKE is secure for any PPT Type One adversary $\mathcal{A}$ under the BR model.*

Proof. The security analysis is performed with a sequence of games $G_{1,l}$ for $0 \leq l \leq 4$. It starts with the real security game, between an adversary $\mathcal{A}$ and a simulator $\mathcal{S}$, that models the indistinguishability of the fresh session key. Use *underline* to show the differences between the previous game and its next one. Let $F_{1,l}$ be the event that $\mathcal{A}$ outputs a guess b': $b' = b$ in *Game* $G_{1,l}$, $l = 0, 1, 2, 3, 4$.

Game $G_{1,0}$. This is the original game where the messages are generated honestly. In Game $G_{1,0}$, $\mathcal{S}$ randomly selects $A^*, B^* \leftarrow \{1, \ldots, N\}$ and $s_{A^*}, s_{B^*} \leftarrow \{1, \ldots, m\}$, and hopes the adversary will choose $sid^* = (\Pi, I, A^*, B^*, R_{A^*}, (R_{B^*}, \omega_{B^*}))$ as the test session, where R_{A^*} is output by s_{A^*}th session of party A^*, and R_{B^*} is output by s_{B^*}th session of party B^* activated by a $Send_1(\Pi, R, B^*, A^*, R_{A^*})$. $\mathcal{S}$ selects $M \leftarrow Z_q^{n \times n}$ at random, honestly generates static public keys for all parties by randomly choosing $s_A, e_A \leftarrow \mathcal{D}_{\mathbb{Z}^n, \alpha q}$, and simulates the attack environment for $\mathcal{A}$. Specifically, $\mathcal{S}$ keeps one table L for random oracle h and responds to queries of $\mathcal{A}$.

(i) h(in) queries: if there is no tuple (in, out) in L, it randomly selects an element out $\in \{0, 1\}^k$ and adds (in, out) to L list. At last it returns out to $\mathcal{A}$.

(ii) $Send_0(\Pi, I, \overline{A}, \overline{B})$: $\mathcal{A}$ initiates a new session of $\overline{A}$ with intended partner $\overline{B}$, $\mathcal{S}$ randomly selects $r_A, e_1 \leftarrow \mathcal{D}_{\mathbb{Z}^n,\alpha q}$ and sends $R_A = Mr_A + 2e_1 \bmod q \in Z_q^n$ to $\mathcal{A}$ on behalf of Alice ($\overline{A}$).

(iii) $Send_1(\Pi, R, \overline{B}, \overline{A}, R_A)$: $\mathcal{S}$ randomly selects $r_B, e_2 \leftarrow \mathcal{D}_{\mathbb{Z}^n,\alpha q}$ and computes $R_B = r_B^t M + 2e_2^t \bmod q \in Z_q^n$, k_B, ω_B, and sk_B according to the protocol. Finally, send (R_B, ω_B) to $\mathcal{A}$ on behalf of Bob.

(iv) $Send_2(\Pi, I, \overline{A}, \overline{B}, R_A, (R_B, \omega_B))$: $\mathcal{S}$ computes k_A, sk_A by using r_A, s_A according to the protocol.

(v) *SessionKeyReveal(sid)*: let $sid = (\Pi, *, \overline{A}, *, *, *, *)$; $\mathcal{S}$ returns sk_A if session key of *sid* has been produced.

(vi) $Corrupt(\overline{A})$: return Alice's static secret key s_A to $\mathcal{A}$.

(vii) *Test(sid)*: let $sid = (\Pi, I, \overline{A}, \overline{B}, R_A, (R_B, \omega_B))$; if $(\overline{A}, \overline{B}) \neq (A^*, B^*)$, or R_A and R_B are not output by the s_{A^*}th session of A^* and the s_{B^*}th session of B^*, respectively, $\mathcal{S}$ stops. Otherwise, $\mathcal{S}$ randomly selects $b \leftarrow \{0, 1\}$, and $sk'_A \leftarrow \{0, 1\}^k$. If $b = 0$, $\mathcal{S}$ returns sk'_A; else it returns the real session key sk_A of *sid*.

Analysis of $G_{1,0}$. In this game, $\mathcal{S}$ randomly selects $A^*, B^* \leftarrow \{1, \ldots, N\}$ and $s_{A^*}, s_{B^*} \leftarrow \{1, \ldots, m\}$ independently from the view of $\mathcal{A}$. Hence, the probability that S will not stop in $G_{1,0}$ is at least $1/(m^2 \cdot N^2)$.

Game $G_{1,1}$. $\mathcal{S}$ first computes $\underline{R'_B = r'^t_B M + 2e'^t_2 \bmod q}$, where $r'_B, e'_2 \leftarrow \mathcal{D}_{\mathbb{Z}^n,\alpha q}$ randomly. Then, it acts almost the same as in $G_{1,0}$ except for the case below.

(i) $Send_1(\Pi, R, \overline{B}, \overline{A}, R_A)$. If $(\overline{A}, \overline{B}) \neq (A^*, B^*)$, or it is not the s_{B^*}th session of B^*, $\mathcal{S}$ responds to the query as in *Game* $G_{1,0}$. Otherwise, $\mathcal{S}$ randomly selects $d \leftarrow \mathcal{D}_{\mathbb{Z},\alpha q}$ and computes $\underline{R_B = R'_B - d \cdot p_B \bmod q}$. Then $\mathcal{S}$ computes $\underline{k_B = r'^t_B \cdot p_A + r'^t_B \cdot R_A \bmod q}$:

$$\begin{aligned} k_B &= r'^t_B \cdot p_A + r'^t_B \cdot R_A \\ &= r'^t_B \cdot (Ms_A + 2e_A) + r'^t_B \cdot (Mr_A + 2e_1) \\ &= r'^t_B \cdot Ms_A + r'^t_B \cdot Mr_A + r'^t_B \cdot 2e_A + 2r'^t_B \cdot e_1 \\ &\quad \bmod q. \end{aligned} \tag{13}$$

Finally, it honestly computes ω_B, sk_B according to the protocol and sends (R_B, ω_B) to $\mathcal{A}$.

Analysis of $G_{1,1}$. Since $(M, R'_B = r'^t_B \cdot M + 2e'^t_2)$ is a LWE tuple with randomly $r'_B, e'_2 \leftarrow \mathcal{D}_{\mathbb{Z}^n,\alpha q}$, then R'_B is computationally indistinguishable from uniform distribution over Z_q^n; thus the probability that $\mathcal{A}$ guesses the correct $R_B = R'_B - d \cdot p_B$ before is negligible. Since $p_B = s_B^t M + 2e_B^t \bmod q$, $s_B, e_B \leftarrow \mathcal{D}_{\mathbb{Z}^n,\alpha q}$, we have

$$R_B = R'_B - d \cdot p_B = \left(r'^t_B - ds_B^t\right) M + 2\left(e'^t_2 - de_B^t\right) \bmod q. \tag{14}$$

By Lemma 1, the norm of each entry in both $r'^t_B - ds_B^t$ and $e'^t_2 - de_B^t$ is at most $s\sqrt{n}$; thus both $r'^t_B - ds_B^t$ and $e'^t_2 - de_B^t$ have distribution negligibly close to $\mathcal{D}_{\mathbb{Z}^n,\alpha q}$. This implies that the distribution of R_B in *Game* $G_{1,1}$ is statistically close to that in *Game* $G_{1,0}$. Thus, under the HNF-LWE$_{q,n,\alpha}$ assumption, we have $\Pr[F_{1,1}] = \Pr[F_{1,0}] - \mathrm{negl}(n)$.

Game $G_{1,2}$. $\mathcal{S}$ computes $\underline{R'_A = Mr'_A + 2e'_1 \bmod q}$, where it randomly selects $r'_A, e'_1 \leftarrow \mathcal{D}_{\mathbb{Z}^n,\alpha q}$. Then it acts almost the same as in $G_{1,1}$ apart from the following cases.

(i) $Send_0(\Pi, I, \overline{A}, \overline{B})$. If $(\overline{A}, \overline{B}) \neq (A^*, B^*)$, or it is not the s_{A^*}th session of A^*, $\mathcal{S}$ responds to the query as in *Game* $G_{1,1}$. Otherwise, $\mathcal{S}$ randomly selects $c \leftarrow \mathcal{D}_{\mathbb{Z},\alpha q}$ and computes $\underline{R_A = R'_A - c \cdot p_A \bmod q}$. Finally, it sends R_A to $\mathcal{A}$.

(ii) $Send_2(\Pi, I, \overline{A}, \overline{B}, R_A, (R_B, \omega_B))$. If $(\overline{A}, \overline{B}) \neq (A^*, B^*)$ or it is not the s_{A^*}th of A^*, $\mathcal{S}$ responds to the query as in *Game* $G_{1,1}$. Otherwise, if (R_B, ω_B) is output by the s_{B^*}th session of party B^*, let sk_B be the session key of session $sid = (\Pi, R, \overline{B}, \overline{A}, R_A, (R_B, \omega_B))$, $\mathcal{S}$ sets $\underline{sk_A = sk_B}$. Else, $\mathcal{S}$ computes $\underline{k_A = p_B \cdot r'_A + R_B \cdot r'_A \bmod q}$. At last, it honestly computes $\underline{sk_A}$ according to the protocol.

$$\begin{aligned} k_A &= p_B \cdot r'_A + R_B \cdot r'_A \\ &= \left(s_B^t M + 2e_B^t\right) \cdot r'_A + \left(r_B^t M + 2e_2^t\right) \cdot r'_A \\ &= s_B^t M \cdot r'_A + r_B^t M \cdot r'_A + 2e_B^t \cdot r'_A + 2e_2^t \cdot r'_A \\ &\quad \bmod q. \end{aligned} \tag{15}$$

Analysis of $G_{1,2}$. The proof of the distribution of R_A being statistically close to that in *Game* $G_{1,1}$ is the same as the proof of *Analysis of* $G_{1,1}$; as a result, the probability that $\mathcal{S}$ stops in $G_{1,2}$ is negligibly close to that of $G_{1,1}$. Under the LWE$_{q,n,\alpha}$ assumption, we have

$$\Pr[F_{1,2}] = \Pr[F_{1,1}] - \mathrm{negl}(n). \tag{16}$$

Games $G_{1,3}$. $\mathcal{S}$ randomly chooses $\underline{R'_A \leftarrow Z_q^n}$ and acts almost the same as in $G_{1,2}$ apart from the following case.

(i) $Send_2(\Pi, I, \overline{A}, \overline{B}, R_A, (R_B, \omega_B))$. If $(\overline{A}, \overline{B}) \neq (A^*, B^*)$, or it is not the s_{A^*}th session of party A^*, or (R_B, ω_B) is output by the s_{B^*}th session of party B^*, $\mathcal{S}$ acts the same as in *Game* $G_{1,2}$. Else, it randomly selects $\underline{sk_A \leftarrow \{0, 1\}^k}$ as the session key.

We discuss the differences between $G_{1,2}$ and $G_{1,3}$. The real session key sk_A in *Game* $G_{1,2}$ is changed to a randomly chosen one in *Game* $G_{1,3}$, where (R_B, ω'_B) $(\omega_B \neq \omega'_B)$ is output by the s_{B^*}th session of party B^*. Fortunately, the adversary cannot notice the difference if he does not query h with exact σ_A, since h is a random oracle. To prove it conveniently, denote $Q_{1,l}$ by the event that in *Game* $G_{1,l}$ $\mathcal{A}$ can query h with σ_A for the s_{A^*}th session of A^*, when (R_B, ω'_B) $(\omega_B \neq \omega'_B)$ is output by the s_{B^*}th session of B^*, where $l = 2, 3, 4$.

Analysis of $G_{1,3}$. Since h is a random oracle, the event $Q_{1,2}$ is independent from the distribution of sk_A. Namely, no matter whether or not $\mathscr{A}$ obtains sk_A, $P[Q_{1,2}]$ is identical, which is suitable for $\Pr[Q_{1,3}]$. Especially, under the LWE assumption, the public information in $G_{1,2}$ and $G_{1,3}$ is computationally indistinguishable, so that $\Pr[Q_{1,3}] = \Pr[Q_{1,2}] - \mathrm{negl}(n)$. In addition, if $\Pr[Q_{1,i}]$, $l = 2, 3$ does not happen, the distribution of sk_A is identical in $G_{1,2}$ and $G_{1,3}$. Namely, $\Pr[F_{1,3} \mid \neg Q_{1,3}] = \Pr[F_{1,2} \mid \neg Q_{1,2}] - \mathrm{negl}(n)$.

Game $G_{1,4}$. $\mathcal{S}$ randomly selects $R'_B \leftarrow Z_q^n$ and acts almost the same as in $G_{1,3}$ except for the following case.

(i) $Send_1(\Pi, R, B, A, R_A)$. If $(\overline{A}, \overline{B}) \neq (A^*, B^*)$ or it is not the s_B^*th session of B^*, $\mathcal{S}$ responds to the query as in *Game* $G_{1,3}$. Otherwise, randomly select $d \leftarrow \mathscr{D}_{\mathbb{Z},\alpha q}$ and compute $R_B = R'_B - p_B \cdot d \bmod q$. $\mathcal{S}$ randomly choose $k_B \leftarrow Z_q$ and compute ω_B, σ_B according to the protocol. If $\mathscr{A}$ made a query $h(\overline{A}, \overline{B}, R_A, R_B, \omega_B, \sigma_B)$, $\mathcal{S}$ stops the simulation. Else, it randomly selects $sk_B \leftarrow \{0,1\}^k$ and sets $h(\overline{A}, \overline{B}, R_A, R_B, \omega_B, \sigma_B) = sk_B$. At last, it returns (R_B, ω_B) to $\mathscr{A}$.

Analysis of $G_{1,4}$. Let $(f_1, h_1), (f_2, h_2)$ be two challenge LWE tuples with error distribution (scale by multiplying $t = 2$). Suppose that there exists an adversary that distinguishes *Game* $G_{1,3}$ and *Game* $G_{1,4}$, there must exist a distinguisher $\mathscr{D}$ that can solve the LWE problem. In particular, $\mathscr{D}$ first sets $M = f_1$, $R'_A = f_2$, $R'_B = h_1$; then it acts identically as $\mathcal{S}$ in *Game* $G_{1,3}$ except for cases as follows.

(i) $Send_1(\Pi, R, \overline{B}, \overline{A}, R_A)$. If $(\overline{A}, \overline{B}) \neq (A^*, B^*)$ or it is not the s_{B^*}th session of B^*, $\mathscr{D}$ responds to queries as in *Game* $G_{1,3}$. Otherwise, it randomly selects $d \leftarrow \mathscr{D}_{\mathbb{Z},\alpha q}$ and computes $R_B = R'_B - dp_B \bmod q$. Then $\mathscr{D}$ sets $k_B = h_2$, computes ω_B, σ_B according to the protocol, and stops if $\mathscr{A}$ made a h query $h(\overline{B}, \overline{A}, R_A, R_B, \omega_B, \sigma_B)$. Otherwise, it sets $h(\overline{B}, \overline{A}, R_A, R_B, \omega_B, \sigma_B) = sk_B$ with a randomly chosen $sk_B \leftarrow \{0,1\}^k$. At last, it returns (R_B, ω_B) to $\mathscr{A}$.

If $(f_1, h_1), (f_2, h_2)$ are LWE tuples for some secret s', $\mathscr{A}$ is in *Game* $G_{1,3}$; else it is in *Game* $G_{1,4}$. Thus, under the $\mathrm{LWE}_{q,n,\alpha}$ assumption, *Game* $G_{1,3}$ and *Game* $G_{1,4}$ are computationally indistinguishable. Especially,

$$\begin{aligned} \Pr[Q_{1,4}] &= \Pr[Q_{1,3}], \\ \Pr[F_{1,4} \mid \neg Q_{1,4}] &= \Pr[F_{1,3} \mid \neg Q_{1,3}] - \mathrm{negl}(n). \end{aligned} \tag{17}$$

Next, we analyze

$$\begin{aligned} \Pr[Q_{1,4}] &= \mathrm{negl}(n), \\ \Pr[F_{1,4} \mid \neg Q_{1,4}] &= \frac{1}{2} + \mathrm{negl}(n). \end{aligned} \tag{18}$$

Let (R_B, ω_B) be output by the s_{B^*}th session of $\overline{B} = B^*$ and (R_B, ω'_B) be the information that finishes the test session (e.g., the s_{A^*}th session of party $\overline{A} = A^*$). In $G_{1,4}$, k_B is randomly selected from uniform distribution over Z_q which is independent from both public keys and transcripts (except for ω_B). This still holds even if the adversary uses a session key reveal query to obtain sk_B, since sk_B is randomly chosen and h is a random oracle. Let k_A be the element computed by S; according to the protocol, k_A and k_B are sufficiently close; that is, $k_A = k_B + g \bmod q$;

$$\begin{aligned} k_A &= k_B + 2\left(r_B^t \cdot e_A + s_B^t \cdot e_1 - e_B^t \cdot r_A - e_2^t \cdot s_A\right) \\ &\triangleq k_B + g \mod q \end{aligned} \tag{19}$$

for some g with short element. Since both public keys and transcripts (expect for ω_B) are random and independent from k_B, g is also independent from k_B without the adversary's view. Since $\mathrm{mod}_2(k_A, \omega'_B) = \mathrm{mod}_2(k_B + g, \omega'_B)$, then $\sigma'_B = \mathrm{mod}_2(k_A, \omega'_B)$ is also statistically close to $\{0,1\}$ by Lemma 4. Namely, the probability that the adversary query $h(\overline{B}, \overline{A}, R_A, R_B, \omega'_B, \sigma'_B)$ is at most $2^{-n} + \mathrm{negl}(n)$. Thus, $\Pr[Q_{1,4}] = \mathrm{negl}(n)$.

Let (R_B, ω_B) be output by the s_{B^*}th session of party $\overline{B} = B^*$ and (R_B, ω'_B) be the message that can finish the test session (e.g., the s_{A^*}th session of party $\overline{A} = A^*$). Consider two cases.

(ii) $\omega_B = \omega'_B$. In this case, $sk_A = sk_B = h(\overline{A}, \overline{B}, R_A, R_B, \omega_B, \sigma_B)$, where $\sigma_B = \mathrm{mod}_2(k_B, \omega_B)$. Since, in $G_{1,4}$, k_B is randomly chosen from the uniform distribution over Z_q, then σ_B is statistically close to uniform distribution over $\{0,1\}$ from the adversary's point by Lemma 4. Thus the probability that $\mathscr{A}$ has made h query with σ_B is less than $2^{-n} + \mathrm{negl}(n)$.

(iii) $\omega_B \neq \omega'_B$. By assumption that $Q_{1,4}$ does not happen, thus $\mathscr{A}$ will never make a h query with σ_B.

In short, the probability that $\mathscr{A}$ has made a h query with σ_A is negligible since h is a random oracle. If the adversary does not make a query with σ_A exactly, sk'_As distribution is uniform over $\{0,1\}^k$ in the adversary's point. Thus, $\Pr[F_{1,4} \mid \neg Q_{1,4}] = 1/2 + \mathrm{negl}(n)$.

In a word, we get $\Pr[F_{1,0}] = \Pr[F_{1,2}] + \mathrm{negl}(n)$ by Analysis of $G_{1,1}$ and $G_{1,2}$. By Analysis of $G_{1,3}$ and $G_{1,4}$, we have

$$\begin{aligned} \Pr[Q_{1,2}] &= \Pr[Q_{1,4}] + \mathrm{negl}(n) = \mathrm{negl}(n) \\ \Pr[F_{1,2} \mid \neg Q_{1,2}] &= \Pr[F_{1,4} \mid \neg Q_{1,4}] + \mathrm{negl}(n). \end{aligned} \tag{20}$$

By the law of the probability,

$$\begin{aligned} \Pr[F_{1,2}] = \Pr[F_{1,2} \mid \neg Q_{1,2}](1 - \Pr[Q_{1,2}]) \\ + \Pr[F_{1,2} \mid Q_{1,2}] \Pr[Q_{1,2}]. \end{aligned} \tag{21}$$

Thus

$$\Pr[F_{1,2}] = \Pr[F_{1,2} \mid \neg Q_{1,2}] - \mathrm{negl}(n). \tag{22}$$

Combining this with $\Pr[F_{1,4} \mid \neg Q_{1,4}] = 1/2 + \mathrm{negl}(n)$, we obtain

$$\Pr[F_{1,0}] = \Pr[F_{1,2}] + \mathrm{negl}(n) = \frac{1}{2} + \mathrm{negl}(n). \tag{23}$$

This finishes the proof. □

4.1.2. Type Two Adversary. We will prove that our AKE is secure against any *PPT Type Two adversary* $\mathcal{A}$.

Theorem 6. *If the HNF – LWE$_{q,n,\alpha}$ is hard, the proposed AKE is secure against any PPT Type Two adversary $\mathcal{A}$ under the BR model.*

Proof. It proceeds by a sequences of *Games* $G_{2,l}$, $0 \le l \le 6$.

Game $G_{2,0}$. This initial game corresponds to the real attack game in which all the honest players execute. $\mathcal{S}$ randomly selects $A^*, B^* \leftarrow \{1, \ldots, N\}$, $s_{A^*} \leftarrow \{1, \ldots, m\}$, and hopes that the adversary will select $sid^* = (\Pi, I, A^*, B^*, R_{A^*}, (R_{B^*}, \omega_B))$ as the test session, where R_{A^*} is output by the s_{A^*}th session of party A^* with intended party B^* (remark that sid^* has no matching session for *Type Two adversary*). Then, $\mathcal{S}$ randomly selects $M \leftarrow Z_q^{n\times n}$, produces static public keys for all parties (by randomly choosing $s_A, e_A \leftarrow \mathcal{D}_{\mathbb{Z}^n,\alpha q}$), and simulates the security game for $\mathcal{A}$. In particular, $\mathcal{S}$ preserves one table L for random oracle h and responds to queries from $\mathcal{A}$.

(i) h(in) queries: if there is no tuple (in, out) in L list, randomly select out $\in \{0,1\}^k$ and add (in, out) to L list. Then, send out to $\mathcal{A}$.

(ii) $Send_0(\Pi, I, \overline{A}, \overline{B})$: $\mathcal{A}$ initiates one session of $\overline{A}$ with intended partner $\overline{B}$, $\mathcal{S}$ randomly selects $r_A, e_1 \leftarrow \mathcal{D}_{\mathbb{Z}^n,\alpha q}$ and returns $R_A = Mr_A + 2e_1 \bmod q \in Z_q^n$ to $\mathcal{A}$ on behalf of Alice ($\overline{A}$).

(iii) $Send_1(\Pi, R, \overline{B}, \overline{A}, R_A)$. $\mathcal{S}$ randomly selects $r_B, e_2 \leftarrow \mathcal{D}_{\mathbb{Z}^n,\alpha q}$ and computes $R_B = r_B^t M + 2e_2^t \bmod q \in Z_q^n$, k_B, ω_B, and sk_B according to the protocol. Then, send (R_B, ω_B) to $\mathcal{A}$ on behalf of Bob.

(iv) $Send_2(\Pi, I, \overline{A}, \overline{B}, R_A, (R_B, \omega_B))$: $\mathcal{S}$ computes k_A, sk_A by using r_A, s_A according to the protocol.

(v) *SessionKeyReveal(sid)*: let $sid = (\Pi, *, \overline{A}, *, *, *, *)$; $\mathcal{S}$ returns sk_A once session key of *sid* has been produced.

(vi) $Corrupt(\overline{A})$: return Alice's static secret key s_A to $\mathcal{A}$.

(vii) *Test(sid)*: let $sid = (\Pi, I, \overline{A}, \overline{B}, R_A, (R_B, \omega_B))$; if $(\overline{A}, \overline{B}) \neq (A^*, B^*)$, or R_A and R_B are not output by the s_{A^*}th session of A^* and the s_{B^*}th session of B^*, respectively, $\mathcal{S}$ stops. Otherwise, $\mathcal{S}$ randomly selects $b \leftarrow \{0,1\}$ and $sk'_A \leftarrow \{0,1\}^k$. If $b = 0$, $\mathcal{S}$ returns sk'_A; else it returns the real session sk_A of *sid*.

Analysis of $G_{2,0}$. In this game, $\mathcal{S}$ randomly selects $A^*, B^* \leftarrow \{1, \ldots, N\}$ and $s_{A^*} \leftarrow \{1, \ldots, m\}$ independently from $\mathcal{A}$'s view. Thus, the probability that S will not stop in $G_{2,0}$ is at least $1/(m \cdot N^2)$.

Game $G_{2,1}$. $\mathcal{S}$ acts identically as in $G_{2,0}$ except for the following case.

(i) $Send_0(\Pi, I, \overline{A}, \overline{B})$. If $\overline{A} \neq A^*$, $\mathcal{S}$ responds to queries as in *Game* $G_{2,0}$. Otherwise, $\mathcal{S}$ computes $\underline{R'_A = Mr'_A + 2e'_1 \bmod q}$, where $r'_A, e'_1 \leftarrow \mathcal{D}_{\mathbb{Z}^n,\alpha q}$. Then, it randomly selects $c \leftarrow \mathcal{D}_{\mathbb{Z},\alpha q}$, computes $\underline{R_A = R'_A - c \cdot p_A \bmod q}$, and sends R_A to $\mathcal{A}$.

(ii) $Send_1(\Pi, R, \overline{B}, \overline{A}, R_A)$. If $\overline{B} \neq B^*$, $\mathcal{S}$ responds to queries as in *Game* $G_{2,0}$. Otherwise, $\mathcal{S}$ computes $\underline{R'_B = r'^t_B M + 2e'^t_2 \bmod q}$, where $r'_B, e'_2 \leftarrow \mathcal{D}_{\mathbb{Z}^n,\alpha q}$. Then, it randomly selects $d \leftarrow \mathcal{D}_{\mathbb{Z},\alpha q}$ and computes $\underline{R_B = R'_B - d \cdot p_B \bmod q}$. $\mathcal{S}$ also computes $\underline{k_B = r'^t_B \cdot p_A + r'^t_B \cdot R_A \bmod q}$:

$$\begin{aligned} k_B &= r'^t_B \cdot p_A + r'^t_B \cdot R_A \\ &= r'^t_B \cdot (Ms_A + 2e_A) + r'^t_B \cdot (Mr_A + 2e_1) \\ &= r'^t_B \cdot Ms_A + r'^t_B \cdot Mr_A + r'^t_B \cdot 2e_A + 2r'^t_B \cdot e_1 \\ &\quad \bmod q. \end{aligned} \tag{24}$$

Finally, it honestly computes ω_B, sk_B following the protocol, and sends (y_B, ω_B) to $\mathcal{A}$.

(iii) $Send_2(\Pi, I, \overline{A}, \overline{B}, R_A, (R_B, \omega_B))$. If $A \neq A^*$, $\mathcal{S}$ responds to queries as in *Game* $G_{2,0}$. Else, let $R_A = R'_A - p_A c \bmod q$ for $R'_A = Mr'_A + 2e'_1 \bmod q$; $\mathcal{S}$ computes $\underline{k_A = (p_B + R_B) \cdot r'_A \bmod q}$. At last, $\mathcal{S}$ computes sk_A according to the protocol.

Analysis of $G_{2,1}$. Let $F_{2,l}$ be the event that $\mathcal{A}$ outputs a guess $b' = b$ in *Game* $G_{2,l}$, $l = 1, 2, 3, 4, 5, 6$. Similar to *Analysis of* $G_{1,1}$, under HNF-LWE$_{q,n,\alpha}$ assumption, then

$$\Pr[F_{2,1}] = \Pr[F_{2,0}] - \mathrm{negl}(n). \tag{25}$$

Game $G_{2,2}$. $\mathcal{S}$ acts almost identically as in $G_{2,1}$, except it replaces B^*'s public key with a uniformly chosen $\underline{p_{B^*} \leftarrow Z_q^n}$.

Analysis of $G_{2,2}$. Note that the only difference between $G_{2,1}$ and $G_{2,2}$ is that $\mathcal{S}$ replaces $p_{B^*} = s_{B^*}^t M + 2e_{B^*}^t$ in $G_{2,1}$ with a randomly chosen p_{B^*} over Z_q^n in $G_{2,2}$; thus an adversary that can distinguish the difference between $G_{2,1}$ and $G_{2,2}$ could solve the LWE$_{q,n,\alpha}$ problem. That implies that if LWE$_{q,n,\alpha}$ is hard, then

$$\Pr[F_{2,2}] = \Pr[F_{2,1}] - \mathrm{negl}(n). \tag{26}$$

Game $G_{2,3}$. $\mathcal{S}$ first computes $\underline{R'_A = Mr'_A + 2e'_1 \bmod q}$, where it randomly chooses $r'_A, e'_1 \leftarrow \mathcal{D}_{\mathbb{Z}^n,\alpha q}$. Then it acts identically as in $G_{2,2}$ except for such cases as follows.

(i) $Send_0(\Pi, I, \overline{A}, \overline{B})$. If $(\overline{A}, \overline{B}) \neq (A^*, B^*)$, or it is not the s_{A^*}th session of A^*, $\mathcal{S}$ responds to queries as in *Game* $G_{2,2}$. Else, $\mathcal{S}$ randomly selects $c \leftarrow \mathcal{D}_{\mathbb{Z},\alpha q}$ and computes $\underline{R_A = R'_A - c \cdot p_A \bmod q}$. Finally, $\mathcal{S}$ returns R_A to $\mathcal{A}$.

(ii) $Send_2(\Pi, I, \overline{A}, \overline{B}, R_A, (R_B, \omega_B))$. If $(\overline{A}, \overline{B}) \neq (A^*, B^*)$ or it is not the s_{A^*}th of A^*, $\mathcal{S}$ responds to the query as in *Game* $G_{2,2}$. Otherwise, $\mathcal{S}$ computes $k_A = p_B \cdot r'_A + R_B \cdot r'_A \bmod q$, where $r'_A \leftarrow \mathcal{D}_{\mathbb{Z}^n,\alpha q}$. Finally, it computes sk_A according to the protocol.

$$\begin{aligned} k_A &= p_B \cdot r'_A + R_B \cdot r'_A \\ &= \left(s_B^t M + 2e_B^t\right) \cdot r'_A + \left(r_B^t M + 2e_2^t\right) \cdot s'_A \\ &= s_B^t M \cdot r'_A + r_B^t M \cdot r'_A + 2e_B^t \cdot r'_A + 2e_2^t \cdot r'_A \\ &\quad \bmod q. \end{aligned} \tag{27}$$

Analysis of $G_{2,3}$. Similar to *Analysis of* $G_{1,1}$, under HNF-$\mathrm{LWE}_{q,n,\alpha}$ assumption, we have

$$\Pr\left[F_{2,3}\right] = \Pr\left[F_{2,2}\right] - \mathrm{negl}(n). \tag{28}$$

Games $G_{2,4}$. $\mathcal{S}$ first computes $h_1 = Mr'_A + 2\widetilde{e}'_1$, $h_2 = p_B r'_A$, where $r'_A, \widetilde{e}'_1 \leftarrow \mathcal{D}_{\mathbb{Z}^n,\alpha q}$. Then it computes $R'_A = h_1 + 2e'_1 = Mr'_A + 2(\widetilde{e}'_1 + e'_1)$ (or set $R'_A = h_1$), where $e'_1 \leftarrow \mathcal{D}_{\mathbb{Z}^n,\alpha,q}$. Finally, it acts identically as in $G_{2,3}$ except for the following cases.

(i) $Send_2(\Pi, I, \overline{A}, \overline{B}, R_A, (R_B, \omega_B))$. If $(\overline{A}, \overline{B}) \neq (A^*, B^*)$, or it is not the s_{A^*}th session of party A^*, $\mathcal{S}$ responds to queries as in *Game* $G_{2,3}$. Else, $\mathcal{S}$ computes

$$k_A = dh_2 + R_B r'_A = ds_B^t M r'_A + r_B^t M r'_A + 2\left(de_B^t r'_A + e_2^t r'_A\right) \bmod q. \tag{29}$$

At last, it computes sk_A according to the protocol.

Analysis of $G_{2,4}$. In *Game* $G_{2,4}$, we have

$$\begin{aligned} R'_A &= Mr'_A + 2\left(\widetilde{e}'_1 + e'_1\right) \quad \bmod q \\ &\left(R'_A = Mr'_A + 2\widetilde{e}'_1 \bmod q\right) \\ k_A &= s_B^t M r'_A + r_B^t M r'_A + 2\left(e_B^t r'_A + e_2^t r'_A\right) \quad \bmod q, \end{aligned} \tag{30}$$

where $\widetilde{e}'_1, e'_1, e_B, e_2, r'_A \leftarrow \mathcal{D}_{\mathbb{Z}^n,\alpha,q}$. By Lemma 2, the distribution of $\widetilde{e}'_1 + e'_1$ is statistically close to $\mathcal{D}_{\mathbb{Z}^n,\alpha,q}$; the distribution of $e_B^t r'_A + e_2^t r'_A$ is statistically close to $\mathcal{D}_{\mathbb{Z},\alpha,q}$. Thus, if $\mathrm{LWE}_{q,n,\alpha}$ is hard, we get

$$\Pr\left[F_{2,4}\right] = \Pr\left[F_{2,3}\right] - \mathrm{negl}(n). \tag{31}$$

Game $G_{2,5}$. $\mathcal{S}$ acts identically as in $G_{2,4}$ except for the following cases.

(i) $Send_2(\Pi, I, \overline{A}, \overline{B}, R_A, (R_B, \omega_B))$. If $(\overline{A}, \overline{B}) \neq (A^*, B^*)$ or it is not the s_A^*th session of A^*, $\mathcal{S}$ responds to queries as in *Game* $G_{2,4}$. Otherwise, it randomly selects $k_A \leftarrow Z_q$ and computes sk_A according to the protocol.

Analysis of $G_{2,5}$. The only difference between $G_{2,4}$ and $G_{2,5}$ is that $\mathcal{S}$ replaces the real $k_A = h_2 + R_B r'_A$ in *Game* $G_{2,4}$ with a randomly chosen $k_A \in Z_q$ in $G_{2,5}$. Since h is a random oracle, the only difference will not affect the $\mathcal{A}$'s view until it makes a h query with σ_A derived from k_A. Formally, denote $Q_{2,l}$ for $l = 4, 5, 6$ as the event that $\mathcal{A}$ makes a h query with σ_A derived from k_A.

Now, we prove that if $\mathrm{LWE}_{q,n,\alpha}$ is hard, then

$$\begin{aligned} \Pr\left[Q_{2,5}\right] &= \Pr\left[Q_{2,4}\right] \\ \Pr\left[F_{2,4} \mid \neg Q_{2,4}\right] &= \Pr\left[F_{2,5} \mid \neg Q_{2,5}\right] = \frac{1}{2} + \mathrm{negl}(n). \end{aligned} \tag{32}$$

Because h is a random oracle, $Q_{2,4}$ is independent from (sk_A)'s distribution. No matter whether or not $\mathcal{A}$ gets sk_A, $\Pr[Q_{2,4}]$ is identical; $\Pr[Q_{2,5}]$ is so. Besides, if $Q_{2,l}$ for $l = 4, 5$ does not happen, $G_{2,5}$ is actually identical to $G_{2,4}$ in adversary view. In particular, (sk_A)'s distribution is random and uniform over $\{0,1\}^k$; namely, the advantage of $\mathcal{A}$ guessing b is negligible if $Q_{2,5}$ does not happen. This finishes the proof.

If $\Pr[Q_{2,5}] \leq \mathrm{negl}(n)$, this completes the proof. But it is not easy to do so. As a matter of fact, though h_2 is random in the adversary's view under the LWE assumption, we cannot have the fact that $k_A = dh_2 + R_B r'_A$ is random since $R_B r'_A$ is related to h_2. Now we show k_A is random. If we randomly chose another $\widetilde{d} \leftarrow \mathcal{D}_{\mathbb{Z},\alpha,\mathrm{q}}$ and obtain $k'_A = \widetilde{d}h_2 + R_B r'_A$ we have $k'_A = k_A + (\widetilde{d} - d)h_2$. That is, $k'_A - k_A = (\widetilde{d} - d)h_2$. Naturally, if the adversary can distinguish k_A (and k'_A) from a uniformly chosen one, it can distinguish h_2 (which is computationally under the LWE assumption) from a random chosen from Z_q.

Actually, $Q_{2,5}$ will happen with negligible probability according to Lemma 1. Let $sid^* = (\Pi, I, A^*, B^*, R_A, (R_B, \omega_B))$ be the test session. By assumption that $\mathcal{A}$ is a *Type Two* adversary, namely, R_B is not output by B^* by a $Send_1(\Pi, R, B^*, A^*, R_A)$, given $h_1 = Mr'_A + 2\widetilde{e}'_1$, $h_2 = p_B r'_A$ in $G_{2,5}$, denote $k_A = dh_2 + R_B r'_A$ (which is the same as in $G_{2,4}$), where $\widetilde{d} \leftarrow \mathcal{D}_{\mathbb{Z},\alpha,q}$. By our assumption, $\mathcal{A}$ will make a h query with σ_A derived from k_A with probability at least $\Pr[Q_{2,5}]$.

Now, fixing h_1, h_2, r'_A which are all chosen by $\mathcal{S}$ and are independent from the adversary's actions, $\mathcal{S}$ sets $\widetilde{d} \neq d$ by randomly choosing $\widetilde{d}, d \leftarrow \mathcal{D}_{\mathbb{Z},\alpha,q}$ and sets $k'_A = k_A + (\widetilde{d} - d)h_2$. By Lemma 1, $\mathcal{A}$ will utilize R_B to finish the test session and makes a h query with σ'_A derived from k'_A with probability at least $P[Q_{2,5}] \cdot 2^{-n}$. Denote $d - Q_{2,l}$ such that an event for $l = 5, 6$ $\mathcal{A}$ in *Game* $G_{2,l}$ will make σ_A, σ'_A in $\mathcal{A}$'s two runs, where σ_A is derived from k_A in $\mathcal{A}$'s first run and σ'_A is derived from $k'_A = k_A + (\widetilde{d} - d)h_2$ in $\mathcal{A}$'s second run. In particular, $\Pr[d - Q_{2,5}] \geq P[Q_{2,5}] \cdot 2^{-n}$.

Game $G_{2,6}$. $\mathcal{S}$ randomly chooses $h_1 \leftarrow Z_q^n$, $h_2 \leftarrow Z_q$ and acts identically as in *Game* $G_{2,5}$.

Analysis of $G_{2,6}$. Since the only difference between $G_{2,6}$ and $G_{2,5}$ is that $\mathcal{S}$ replaces $h_1 = Mr'_A + 2\tilde{e}'_1$ and $h_2 = p_B r'_A$ with randomly chosen elements in Z_q^n, Z_q, respectively, an adversary that distinguishes the difference between $G_{2,5}$ and $G_{2,6}$ could solve the $\mathrm{LWE}_{q,n,\alpha}$ problem. Hence, under the $\mathrm{HNF\text{-}LWE}_{q,n,\alpha}$ assumption, *Game* $G_{2,6}$ is computationally indistinguishable from *Game* $G_{2,5}$. In particular, we get

$$\Pr\left[d - G_{2,6}\right] = \Pr\left[d - G_{2,5}\right] - \mathrm{negl}\,(n)\,. \tag{33}$$

Besides, we can get $\Pr[d - G_{2,6}] = \mathrm{negl}(n)$. Actually, in *Game* $G_{2,6}$, $\mathcal{S}$ does not really compute k_A and k'_A (it cannot compute the values since h_1 is randomly chosen from Z_q^n and h_2 is randomly chosen from Z_q). Suppose that k_A and k'_A (e.g., the values determined before) are as $\mathcal{A}$'s target values. $k'_A = k_A + (\tilde{d} - d)h_2$ especially holds, since $\mathcal{A}$ cannot efficiently distinguish *Game* $G_{2,6}$ from $G_{2,5}$ as mentioned above. Since h_2 is uniformly distributed over Z_q and independent from $\mathcal{A}$'s view (thus is independent from k_A and k'_A), then $\sigma'_A = \mathrm{mod}_2(k'_A, \omega'_A)$ is statistically close to uniform over $\{0, 1\}$ even when conditioned on $\sigma_A = \mathrm{mod}_2(k_A, \omega_A)$ by Lemma 4 ($(\tilde{d} - d)$ is in invertible with overwhelming probability). Thus, the probability that $\mathcal{A}$ will make a h query with σ'_A is at most $2^{-n} + \mathrm{negl}(n)$. Namely, $\Pr(d - G_{2,6}) \leq 2^{-n} + \mathrm{negl}(n)$, which is negligible in n. Namely, $\Pr[d - G_{2,6}] = \mathrm{negl}(n)$.

Generally speaking, by *Analysis of* $G_{2,6}$, we have $\Pr[d - G_{2,5}] = \mathrm{negl}(n)$, which implies that $\Pr[Q_{2,5}] = \mathrm{negl}(n)$ by the condition that $\Pr[\mathrm{double} - Q_{2,5}] \geq \Pr[Q_{2,5}](1/2^{n^2}))$. Combining this with *Analysis of* $G_{2,5}$, we obtain $\Pr[F_{2,4}] = 1/2 + \mathrm{negl}(n)$. A simple computation shows that $\Pr[F_{2,0}] = 1/2 + \mathrm{negl}(n)$. The proof is completed. □

4.2. Security for the Responder. Now we prove the security when the responder is the owner of the test session. Let $sid^* = (\Pi, R, B^*, A^*, R_{A^*}, (R_{B^*}\omega_{B^*}))$ be the test session; consider three types of adversaries.

Type Three: R_{A^*} is not output by any session of A^* activated by a $Send_0(\Pi, I, A^*, B^*)$.

Type Four: R_A^* is output by a session of A^* activated by a $Send_0(\Pi, I, A^*, B^*)$, but A^* never completes the session, or it completes it with exact R_{B^*}.

Type Five: R_A^* is output by a session of A^* activated by a $Send_0(\Pi, I, A^*, B^*)$, but A^* completes the session with another $R'_B \neq R_B$.

Type Three, *Type Four*, and *Type Five* give a complete partition of all the adversaries that choose sid^* as the test session. Note that if the adversary is a *Type Three* or *Type Five* one, the test session has no matching session. To address wPFS in our security proof, *Type Four* adversary $\mathcal{A}$ is allowed to obtain the static secret keys of parties A^* and B^* by corrupting both parties (a *Type Three* or *Type Five* adversary is not allowed to corrupt party A^* or B^*).

4.2.1. Type Three. Here we prove AKE security against any PPT *Type Three* adversary $\mathcal{A}$.

Theorem 7. *If HNF − LWE*$_{q,n,\alpha}$ *is hard, the proposed AKE is secure against any PPT Type Three adversary* $\mathcal{A}$ *under the BR model.*

Proof. We prove the theorem by a series of games $G_{3,l}$ for $0 \leq l \leq 7$.

Game $G_{3,0}$. $\mathcal{S}$ randomly selects $A^*, B^* \leftarrow \{1, \ldots, N\}$ and $s_{B^*} \leftarrow \{0, \ldots, m\}$ and hopes that the adversary will select $sid^* = (\Pi, R, B^*, A^*, R_{A^*}, (R_{B^*}, \omega_B))$ as the test session, where (R_{B^*}, ω_B) is output by the s_{B^*}th session of party B^* activated by $Send_0(\Pi, R, B^*, A^*, R_{A^*})$ for some party A^*. Then, $\mathcal{S}$ randomly selects $M \leftarrow Z_q^{n\times n}$, honestly generates static public keys for all parties (by randomly choosing $s_A, e_A \leftarrow \mathcal{D}_{\mathbb{Z}^n,\alpha q}$), and simulates security game for $\mathcal{A}$. Specially, $\mathcal{S}$ preserves one table L for random oracle h, and responds to queries from $\mathcal{A}$ as follows.

(i) h(in) queries: if there is no tuple (in, out) in L, randomly select an element out $\in \{0, 1\}^k$ and add (in, out) to L list. Then send *out* to $\mathcal{A}$.

(ii) $Send_0(\Pi, I, \overline{A}, \overline{B})$: $\mathcal{A}$ initiates a new session of $\overline{A}$ with intended partner $\overline{B}$, $\mathcal{S}$ randomly selects $r_A, e_1 \leftarrow \mathcal{D}_{\mathbb{Z}^n,\alpha q}$ and sends $R_A = Mr_A + 2e_1 \bmod q \in Z_q^n$ to $\mathcal{A}$ on behalf of Alice ($\overline{A}$).

(iii) $Send_1(\Pi, R, \overline{B}, \overline{A}, R_A)$: $\mathcal{S}$ randomly selects $r_B, e_2 \leftarrow \mathcal{D}_{\mathbb{Z}^n,\alpha q}$ and computes $R_B = r_B^t M + 2e_2^t \bmod q \in Z_q^n$, k_B, ω_B, and sk_B according to the protocol. At last, send (R_B, ω_B) to $\mathcal{A}$ on behalf of Bob.

(iv) $Send_2(\Pi, I, \overline{A}, \overline{B}, R_A, (R_B, \omega_B))$: $\mathcal{S}$ computes k_A, sk_A by using r_A, s_A according to the protocol.

(v) *SessionKeyReveal(sid)*: let $sid = (\Pi, *, \overline{A}, *, *, *, *)$; $\mathcal{S}$ returns sk_A once session key of sid has been generated.

(vi) $Corrupt(\overline{A})$: return the static secret key s_A of Alice to $\mathcal{A}$.

(vii) *Test(sid)*: let $sid = (\Pi, I, \overline{A}, \overline{B}, R_A, (R_B, \omega_B))$; if $(\overline{A}, \overline{B}) \neq (A^*, B^*)$, or R_A and R_B are not output by the s_{A^*}th session of A^* and the s_{B^*}th session of B^*, respectively, $\mathcal{S}$ stops. Otherwise, $\mathcal{S}$ randomly selects $b \leftarrow \{0, 1\}$, and $sk'_A \leftarrow \{0, 1\}^k$. If $b = 0$, $\mathcal{S}$ returns sk'_A; else it returns the real session sk_A of sid.

Analysis of $G_{3,0}$. In this game, $\mathcal{S}$ randomly chooses $A^*, B^* \leftarrow \{1, \ldots, N\}$ and $s_{B^*} \leftarrow \{1, \ldots, m\}$ without $\mathcal{A}$'s view. Thus, the probability that S will not abort in $G_{3,0}$ is at least $1/(m \cdot N^2)$.

Game $G_{3,1}$. $\mathcal{S}$ acts identically as in $G_{3,0}$ except for the following cases.

(i) $Send_0(\Pi, I, \overline{A}, \overline{B})$. If $\overline{A} \neq A^*$, $\mathcal{S}$ responds to queries as in *Game* $G_{3,0}$. Else, $\mathcal{S}$ computes $R'_A = Mr'_A + 2e'_1 \bmod q$, where $r'_A, e'_1 \leftarrow \mathcal{D}_{\mathbb{Z}^n,\alpha q}$. Then, it randomly selects $c \leftarrow \mathcal{D}_{\mathbb{Z},\alpha q}$, computes $R_A = R'_A - c \cdot p_A \bmod q$, and sends R_A to $\mathcal{A}$.

(ii) $Send_1(\Pi, R, \overline{B}, \overline{A}, R_A)$. If $\overline{B} \neq B^*$, $\mathcal{S}$ responds to the query as in *Game* $G_{3,0}$. Otherwise, $\mathcal{S}$ computes $R'_B = r'^t_B M + 2e'^t_2 \bmod q$, where $r'_B, e'_2 \leftarrow \mathcal{D}_{\mathbb{Z}^n,\alpha q}$. Then, $\mathcal{S}$ randomly selects $d \leftarrow \mathcal{D}_{\mathbb{Z},\alpha q}$ and computes $R_B = R'_B - d \cdot p_B \bmod q$ and $k_B = r'^t_B \cdot p_A + r'^t_B \cdot R_A \bmod q$:

$$\begin{aligned} k_B &= r'^t_B \cdot p_A + r'^t_B \cdot R_A \\ &= r'^t_B \cdot (Ms_A + 2e_A) + r'^t_B \cdot (Mr_A + 2e_1) \\ &= r'^t_B \cdot Ms_A + r'^t_B \cdot Mr_A + r'^t_B \cdot 2e_A + 2r'^t_B \cdot e_1 \\ &\quad \bmod q. \end{aligned} \tag{34}$$

At last, it computes ω_B, sk_B according to the protocol and sends (y_B, ω_B) to $\mathcal{A}$.

(iii) $Send_2(\Pi, I, \overline{A}, \overline{B}, R_A, (R_B, \omega_B))$. If $A \neq A^*$, $\mathcal{S}$ responds to query as in *Game* $G_{3,0}$. Else, it chooses randomly $c \leftarrow \mathcal{D}_{\mathbb{Z},\alpha q}$, let $R_A = R'_A - p_A c \bmod q$ for $R'_A = Mr'_A + 2e'_1 \bmod q$; $\mathcal{S}$ computes $k_A = (p_B + R_B) \cdot r'_A \bmod q$. Finally, $\mathcal{S}$ computes sk_A according to the protocol.

Analysis of $G_{3,1}$. Let $F_{3,l}$ be the event that $\mathcal{A}$ outputs a guess $b' = b$ in *Game* $G_{3,l}$, $l = 2, 3, 4, 5, 6$. Similar to the *Analysis of* $G_{1,1}$, under HNF-LWE$_{q,n,\alpha}$ assumption, we have

$$\Pr[F_{3,1}] = \Pr[F_{3,0}] - \mathrm{negl}(n). \tag{35}$$

Game $G_{3,2}$. $\mathcal{S}$ acts identically as in $G_{3,1}$, except it replaces the public key for the party A^* with a uniformly chosen $p_{A^*} \leftarrow Z_q^n$.

Analysis of $G_{3,2}$. Similar to *Analysis of* $G_{2,3}$ or $G_{1,1}$, under HNF-LWE$_{q,n,\alpha}$ assumption, then

$$\Pr[F_{3,2}] = \Pr[F_{3,1}] - \mathrm{negl}(n). \tag{36}$$

Game $G_{3,3}$. $\mathcal{S}$ first computes $R'_B = r'^t_B M + 2e'^t_2 \bmod q$, where it randomly chooses $r'_B, e'_2 \leftarrow \mathcal{D}_{\mathbb{Z}^n,\alpha q}$. Then it acts identically as in $G_{3,2}$ except for the following cases.

(i) $Send_1(\Pi, R, \overline{A}, \overline{B}, R_A)$. If $(\overline{A}, \overline{B}) \neq (A^*, B^*)$, or it is not the s_{B^*}th session of B^*, $\mathcal{S}$ responds to the query as in *Game* $G_{3,2}$. Else, $\mathcal{S}$ randomly selects $d \leftarrow \mathcal{D}_{\mathbb{Z},\alpha q}$ and computes $R_B = R'_B - d \cdot p_B \bmod q$ and $k_B = r'^t_B \cdot p_A + r'^t_B \cdot R_A$, where $r'_B \leftarrow \mathcal{D}_{\mathbb{Z}^n,\alpha q}$. At last, it computes sk_B, ω_B according to the protocol and returns (R_B, ω_B) to $\mathcal{A}$.

$$\begin{aligned} k_B &= r'^t_B \cdot p_A + r'^t_B \cdot R_A \\ &= r'^t_B \cdot (Ms_A + 2e_A) + r'^t_B \cdot (Mr_A + 2e_1) \\ &= r'^t_B \cdot Ms_A + r'^t_B \cdot Mr_A + 2r'^t_B e_A + 2r'^t_B \cdot e_1 \\ &\quad \bmod q. \end{aligned} \tag{37}$$

Analysis of $G_{3,3}$. Similar to Analysis of $G_{1,1}$, under LWE$_{q,n,\alpha}$ assumption, then

$$\Pr[F_{3,3}] = \Pr[F_{3,2}] - \mathrm{negl}(n). \tag{38}$$

Games $G_{3,4}$. $\mathcal{S}$ first computes $h_1 = r'^t_B M + 2\tilde{e}'^t_2$, $h_2 = r'^t_B p_A$, where $r'_B, \tilde{e}'_2 \leftarrow \mathcal{D}_{\mathbb{Z}^n,\alpha q}$. Then it computes $R'_B = h_1 + 2e'^t_2 = r'^t_B M + 2(\tilde{e}'^t_2 + e'^t_2)$ (or set $R'_B = h_1$), where $e'_2 \leftarrow \mathcal{D}_{\mathbb{Z}^n,\alpha,q}$. Finally, it acts identically as in $G_{3,3}$ except for the following cases.

(i) $Send_1(\Pi, R, \overline{B}, \overline{A}, R_A)$. If $(\overline{A}, \overline{B}) \neq (A^*, B^*)$, or it is not the s_{B^*}th session of party B^*, $\mathcal{S}$ responds to queries as in *Game* $G_{3,3}$. Otherwise, $\mathcal{S}$ randomly selects $c \leftarrow \mathcal{D}_{\mathbb{Z},\alpha,q}$, and computes $R_B = R'_B - d \cdot p_B \bmod q$,

$$k_B = ch_2 + r'^t_B R_A = cr^t_B Ms_A + r'^t_B Mr_A + 2\left(dr'^t_B e_A + r'^t_B e_1\right) \bmod q. \tag{39}$$

At last, it computes ω_B, sk_B according to the protocol and returns (R_B, ω_B) to $\mathcal{A}$.

Analysis of $G_{3,4}$. Similar to Analysis of $G_{1,1}$, under LWE$_{q,n,\alpha}$ assumption, then

$$\Pr[F_{3,4}] = \Pr[F_{3,3}] - \mathrm{negl}(n). \tag{40}$$

Game $G_{3,5}$. $\mathcal{S}$ acts identically as in $G_{3,4}$ apart from the following cases.

(i) $Send_1(\Pi, R, \overline{B}, \overline{A}, R_A)$. If $(\overline{A}, \overline{B}) \neq (A^*, B^*)$ or it is not the s^*_B – th session of B^*, $\mathcal{S}$ responds to the query as in *Game* $G_{3,4}$. Otherwise, $\mathcal{S}$ randomly selects $d \leftarrow \mathcal{D}_{\mathbb{Z},\alpha,q}$ and computes $R_B = R'_B - d \cdot p_B \bmod q$. Then, it randomly selects $k_B \leftarrow Z_q$ and computes ω_B, σ_B as described in protocol. If $\mathcal{A}$ has made a h query $h(\overline{A}, \overline{B}, R_A, R_B, \omega_B, \sigma_B)$, $\mathcal{S}$ stops. Otherwise, it randomly selects $sk_B \leftarrow \{0,1\}^k$ and sets $h(\overline{A}, \overline{B}, R_A, R_B, \omega_B, \sigma_B) = sk_B$. At last, it returns (R_B, ω_B) to $\mathcal{A}$.

Analysis of $G_{3,5}$. The only difference between $G_{3,4}$ and $G_{3,5}$ is that $\mathcal{S}$ replace the real key $k_B = ch_2 + r_B'^t R_A = cr_B^t M s_A + r_B'^t M r_A + 2(cr_B'^t e_A + r_B'^t e_1) \bmod q$ in Game $G_{3,4}$ with a randomly chosen $k_B \in Z_q$ in Game $G_{3,5}$. Since h is a random oracle, the only difference cannot affect $\mathcal{A}$'s view until it makes a h query with σ_B derived from k_B. Denote $Q_{3,l}$ for $l = 4, 5, 6$ as the event where $\mathcal{A}$ makes a h query with σ_B derived from k_B.

Now, we prove

$$\begin{aligned} \Pr\left[Q_{3,4}\right] &= \Pr\left[Q_{3,5}\right] \\ \Pr\left[F_{3,4} \mid \neg Q_{3,4}\right] &= \Pr\left[F_{3,5} \mid \neg Q_{3,5}\right] = \frac{1}{2} + \operatorname{negl}(n). \end{aligned} \tag{41}$$

Because h is a random oracle, $Q_{3,4}$ is independent from (sk_B)'s distribution. No matter whether or not $\mathcal{A}$ gets sk_A, $\Pr[Q_{3,4}]$ is identical; $\Pr[Q_{3,5}]$ is so. Besides, if $Q_{3,l}$ for $l = 4, 5$ does not happen, $G_{3,5}$ is actually identical as $G_{3,4}$ in adversary view. In particular, (sk_B)'s distribution is random and uniform over $\{0,1\}^k$; namely, the advantage of $\mathcal{A}$ guessing b is negligible if $Q_{3,5}$ does not happen. This completes the proof.

Likewise, let $sid^* = (\Pi, R, B^*, A^*, R_A, (R_B, \omega_B))$ be the test session. By assumption that $\mathcal{A}$ is a *Type Three* adversary, namely, R_A is not output by party A^*, given $h_1 = r_B'^t M + 2\tilde{e}_2'^t$, $h_2 = r_B'^t p_A$ in $G_{3,5}$, denote $k_B = ch_2 + r_B'^t R_A$ (which is the same as in $G_{3,4}$), where $c \leftarrow \mathcal{D}_{\mathbb{Z},\alpha,q}$. By our assumption, $\mathcal{A}$ will make a h query with σ_B derived from k_B with probability at least $\Pr[Q_{3,5}]$.

Now, fixing h_1, h_2, r_B' which are all chosen by $\mathcal{S}$ and are independent from the adversary's actions, $\mathcal{S}$ sets $\tilde{c} \neq c$ by randomly choosing $\tilde{c} \leftarrow \mathcal{D}_{\mathbb{Z},\alpha,q}$, and sets $k_B' = \tilde{c}h + r_B'^t R_A = k_B + (\tilde{c} - c)h_2$. By Lemma 1, $\mathcal{A}$ will utilize R_A in the test session and makes a h query with σ_B' derived from k_B' with probability at least $P[Q_{3,5}] \cdot 2^{-n}$. Denote $d - Q_{3,l}$ as an event for $l = 5, 6$; $\mathcal{A}$ in *Game* $G_{3,l}$ will make σ_B, σ_B' in $\mathcal{A}$'s two runs, where σ_B is derived from k_B in $\mathcal{A}$'s first run and σ_B' is derived from $k_B' = k_B + (\tilde{c} - c)h_2$ in $\mathcal{A}$'s second run. In particular, $\Pr[d - Q_{3,5}] \geq P[Q_{3,5}] \cdot 2^{-n}$.

Game $G_{3,6}$. $\mathcal{S}$ randomly chooses $h_1 \leftarrow Z_q^n$, $h_2 \leftarrow Z_q$ and acts identically as in *Game* $G_{3,5}$.

Analysis of $G_{3,6}$. On the one hand, the difference between $G_{3,5}$ and $G_{3,6}$ is that $\mathcal{S}$ replaces $h_1 = r_B'^t M + 2\tilde{e}_2'^t$, $h_2 = r_B'^t p_A$ with randomly chosen elements in Z_q^n, Z_q, respectively; an adversary that distinguishes the difference between $G_{3,5}$ and $G_{3,6}$ could solve the HNF-LWE$_{q,n,\alpha}$ problem. Under the HNF-LWE$_{q,n,\alpha}$ assumption, *Game* $G_{3,6}$ is computationally indistinguishable from *Game* $G_{3,5}$. In particular,

$$\Pr\left[d - G_{3,6}\right] = \Pr\left[d - G_{3,5}\right] - \operatorname{negl}(n). \tag{42}$$

On the other hand, in *Game* $G_{3,6}$, $\mathcal{S}$ does not really compute k_B and k_B' (it cannot compute them since h_1 is randomly chosen from Z_q^n and h_2 is randomly chosen from Z_q). Suppose that k_B and k_B' (e.g., the values determined before) are as $\mathcal{A}$'s target values. $k_B' = k_B + (\tilde{c} - c)h_2$ especially holds, since $\mathcal{A}$ cannot distinguish *Game* $G_{3,6}$ from $G_{3,5}$ as mentioned above. Since h_2 is uniformly distributed over Z_q and independent from $\mathcal{A}$'s view (thus is independent from k_B and k_B'), then $\sigma_B' = \bmod_2(k_B', \omega_B')$ is statistically close to uniform over $\{0,1\}$ even when conditioned on $\sigma_B = \bmod_2(k_B, \omega_B)$ by Lemma 4 ($(\tilde{c} - c)$ is in invertible with overwhelming probability). Thus, the probability that $\mathcal{A}$ will make a h query with σ_B' is at most $2^{-n} + \operatorname{negl}(n)$. Namely, $\Pr(d - G_{3,6}) \leq 2^{-n} + \operatorname{negl}(n)$ which is negligible in n. As a result, $\Pr[d - G_{3,6}] = \operatorname{negl}(n)$.

Generally speaking, by *Analysis of* $G_{3,6}$, $\Pr[d - Q_{3,5}] = \operatorname{negl}(n)$, which implies that $\Pr[Q_{3,5}] = \operatorname{negl}(n)$ by the condition that $\Pr[d - Q_{3,5}] \geq \Pr[Q_{3,5}] \cdot 2^{-n}$. Combining this with *Analysis of* $G_{3,5}$, we get $\Pr[F_{3,4}] = 1/2 + \operatorname{negl}(n)$. A simple computation shows that $\Pr[F_{3,0}] = 1/2 + \operatorname{negl}(n)$. The proof is completed. □

4.2.2. Type Four Adversary

Theorem 8. *If HNF − LWE$_{q,n,\alpha}$ is hard, the proposed AKE is secure against any PPT Type Four adversary $\mathcal{A}$ under the BR model.*

Proof. We prove it by a series of Game $G_{4,l}$ for $0 \leq l \leq 4$.

Game $G_{4,0}$. $\mathcal{S}$ randomly selects $A^*, B^* \leftarrow \{1, \ldots, N\}$ and $s_{A^*}, s_{B^*} \leftarrow \{0, \ldots, m\}$ and hopes that the adversary will select $sid^* = (\Pi, R, B^*, A^*, R_{A^*}, (R_{B^*}, \omega_{B^*}))$ as the test session, where R_{A^*} is output by the s_{A^*}th session of party A^*, (R_{B^*}, ω_{B^*}) is output by the s_{B^*}th session of party B^* activated by $Send_1(\Pi, R, B^*, A^*, R_{A^*})$. Then, $\mathcal{S}$ randomly selects $M \leftarrow Z_q^{n \times n}$, generates static public keys for all parties (by randomly choosing $s_A, e_A \leftarrow \mathcal{D}_{\mathbb{Z}^n,\alpha q}$), and simulates security game for $\mathcal{A}$. In particular, $\mathcal{S}$ preserves one table L for random oracle h, and responds the queries from $\mathcal{A}$ in the following.

(i) h(in) queries: if there is no tuple (in, out) in L, randomly select an element out $\in \{0,1\}^k$ and add (in, out) to L list. Then send out to $\mathcal{A}$.

(ii) $Send_0(\Pi, I, \overline{A}, \overline{B})$: $\mathcal{A}$ initiates a session of $\overline{A}$ with intended partner $\overline{B}$; $\mathcal{S}$ randomly selects $r_A, e_1 \leftarrow \mathcal{D}_{\mathbb{Z}^n,\alpha q}$ and sends $R_A = Mr_A + 2e_1 \bmod q \in Z_q^n$ to $\mathcal{A}$ on behalf of Alice ($\overline{A}$).

(iii) $Send_1(\Pi, R, \overline{B}, \overline{A}, R_A)$: $\mathcal{S}$ randomly selects $r_B, e_2 \leftarrow \mathcal{D}_{\mathbb{Z}^n,\alpha q}$ and computes $R_B = r_B^t M + 2e_2^t \bmod q \in Z_q^n$, k_B, ω_B, and sk_B according to protocol. At last, send (R_B, ω_B) to $\mathcal{A}$ on behalf of Bob.

(iv) $Send_2(\Pi, I, \overline{A}, \overline{B}, R_A, (R_B, \omega_B))$: $\mathcal{S}$ computes k_A, sk_A with r_A, s_A according to the protocol.

(v) *SessionKeyReveal(sid):* let $sid = (\Pi, *, \overline{A}, *, *, *, *)$; $\mathcal{S}$ returns sk_A if session key of sid has been generated.

(vi) $Corrupt(\overline{A})$: send the static secret key s_A of Alice to $\mathcal{A}$.

(vii) *Test(sid)*: let $sid = (\Pi, I, \overline{A}, \overline{B}, R_A, (R_B, \omega_B))$; if $(\overline{A}, \overline{B}) \neq (A^*, B^*)$, or R_A and R_B are not output by the s_{A^*}th session of A^* and the s_{B^*}th session of B^*, respectively, $\mathcal{S}$

stops. Otherwise, $\mathcal{S}$ randomly selects $b \leftarrow \{0,1\}$, and $sk'_A \leftarrow \{0,1\}^k$. If $b = 0$, $\mathcal{S}$ returns sk'_A; else it returns the real session key sk_A of sid.

Analysis of $G_{4,0}$. In this game, $\mathcal{S}$ randomly selects $A^*, B^* \leftarrow \{1,\ldots,N\}$ and $s_{A^*}, s_{B^*} \leftarrow \{1,\ldots,m\}$ independent from $\mathcal{A}$'s view. Thus, the probability that S will not stop in $G_{4,0}$ is at least $1/(m^2 \cdot N^2)$.

Let $F_{4,l}$ be the event that $\mathcal{A}$ outputs a guess $b' = b$ in *Game* $G_{4,l}$, $l = 1,2,3,4$.

Game $G_{4,1}$. S first computes $\underline{R'_B = r'^t_B M + 2e'_2 \bmod q}$, where $r'_B, e'_2 \leftarrow \mathcal{D}_{\mathbb{Z}^n,\alpha q}$. Then, it acts identically as in $G_{4,0}$ except for the following cases.

(i) $Send_1(\Pi, R, \overline{B}, \overline{A}, R_A)$. If $(\overline{A}, \overline{B}) \neq (A^*, B^*)$, or it is not the s_{B^*}th session of B^*, $\mathcal{S}$ responds to the query as in *Game* $G_{4,0}$. Otherwise, $\mathcal{S}$ randomly selects $d \leftarrow \mathcal{D}_{\mathbb{Z},\alpha q}$ and computes $\underline{R_B = R'_B - d \cdot p_B \bmod q}$. Then, $\mathcal{S}$ also computes $\underline{k_B = r'^t_B \cdot p_A + r'^t_B \cdot R_A \bmod q}$:

$$\begin{aligned} k_B &= r'^t_B \cdot p_A + r'^t_B \cdot R_A \\ &= r'^t_B \cdot (Ms_A + 2e_A) + r'^t_B \cdot (Mr_A + 2e_1) \\ &= r'^t_B \cdot Ms_A + r'^t_B \cdot Mr_A + r'^t_B \cdot 2e_A + 2r'^t_B \cdot e_1 \\ &\quad \bmod q. \end{aligned} \tag{43}$$

Finally, it honestly computes ω_B, sk_B following the protocol and returns (y_B, ω_B) to $\mathcal{A}$.

Analysis of $G_{4,1}$. Similar to *Analysis of* $G_{1,1}$, under HNF-LWE$_{q,n,\alpha}$ assumption, we have

$$\Pr[F_{4,1}] = \Pr[F_{4,0}] - \mathrm{negl}(n). \tag{44}$$

Game $G_{4,2}$. $\mathcal{S}$ first computes $\underline{R'_A = Mr'_A + 2e'_1 \bmod q}$, where $r'_A, e'_1 \leftarrow \mathcal{D}_{\mathbb{Z}^n,\alpha q}$. Then, $\mathcal{S}$ acts identically as in $G_{4,1}$, apart from the following cases.

(i) $Send_0(\Pi, I, \overline{A}, \overline{B})$. If $(\overline{A}, \overline{B}) \neq (A^*, B^*)$, or it is not the s_{A^*}th session of A^*, $\mathcal{S}$ responds to the query as in *Game* $G_{4,1}$. Else, it randomly selects $c \leftarrow \mathcal{D}_{\mathbb{Z},\alpha q}$, computes $\underline{R_A = R'_A - c \cdot p_A \bmod q}$, and sends R_A to $\mathcal{A}$.

(ii) $Send_2(\Pi, I, \overline{A}, \overline{B}, R_A, (R_B, \omega_B))$. If $(\overline{A}, \overline{B}) \neq (A^*, B^*)$, or it is not the s_{A^*}th session of A^*, $\mathcal{S}$ responds to queries as in *Game* $G_{4,1}$. Otherwise, (R_B, ω_B) is output by the s_{B^*}th session of party B^*. Let sk_B be the session key of session $sid = (\Pi, R, \overline{B}, \overline{A}, R_A, (R_B, \omega_B))$; $\mathcal{S}$ sets $\underline{sk_A = sk_B}$. Otherwise, it computes $\underline{k_A = (p_B + R_B) \cdot r'_A \bmod q}$. At last, $\mathcal{S}$ computes sk_A according to the protocol.

Analysis of $G_{4,2}$. Similar to *Analysis of* $G_{1,2}$, under HNF-LWE$_{q,n,\alpha}$ assumption, we have

$$\Pr[F_{4,2}] = \Pr[F_{4,1}] - \mathrm{negl}(n). \tag{45}$$

Game $G_{4,3}$. $\mathcal{S}$ first randomly selects $\underline{R'_A \leftarrow Z_q^n}$. Then it acts identically as in $G_{4,2}$ except in the following cases.

$Send_2(\Pi, I, \overline{A}, \overline{B}, R_A, (R_B, \omega_B))$. If $(\overline{A}, \overline{B}) \neq (A^*, B^*)$, or it is not the s_{A^*}th session of A^*, or (R_B, ω_B) is output by the s_{B^*}th session of party B^*, $\mathcal{S}$ answers the query as in *Game* $G_{4,2}$. Else, it randomly selects $\underline{sk_A \leftarrow \{0,1\}^k}$ as the session key.

Analysis of $G_{4,3}$. Denote $Q_{4,l}$ as the event where in *Game* $G_{4,l}$ for $l = 2,3,4$, $\mathcal{A}$ makes a h query with σ_A for the s_{A^*}th session of party A^*, when (R_B, ω'_B) is output by the s_{B^*}th session of party B^* but $\omega_B \neq \omega'_B$.

Similar to *Analysis of* $G_{1,3}$, under HNF-LWE$_{q,n,\alpha}$ assumption, we have

$$\begin{aligned} \Pr[Q_{4,3}] &= \Pr[Q_{4,2}] - \mathrm{negl}(n) \\ \Pr[F_{4,3} \mid \neg Q_{4,3}] &= \Pr[F_{4,2} \mid \neg Q_{4,2}] - \mathrm{negl}(n). \end{aligned} \tag{46}$$

Game $G_{4,4}$. $\mathcal{S}$ randomly selects $\underline{R'_B \leftarrow Z_q^n}$ and acts almost identically in $G_{4,3}$ except for the following cases.

(i) $Send_1(\Pi, R, \overline{B}, \overline{A}, R_A)$. If $(\overline{A}, \overline{B}) \neq (A^*, B^*)$ or it is not the s_B^*th session of B^*, $\mathcal{S}$ responds to the query as in *Game* $G_{4,3}$. Otherwise, $\mathcal{S}$ randomly chooses $d \leftarrow \mathcal{D}_{\mathbb{Z},\alpha,q}$ and computes $\underline{R_B = R'_B - d \cdot p_B \bmod q}$. Then $\mathcal{S}$ randomly selects $\underline{k_B \leftarrow Z_q}$ and computes ω_B, σ_B as described in protocol. If A has made a h query $h(\overline{A}, \overline{B}, R_A, R_B, \omega_B, \sigma_B)$, $\mathcal{S}$ stops. Else, it randomly selects $\underline{sk_B \leftarrow \{0,1\}^k}$ and sets $h(\overline{A}, \overline{B}, R_A, R_B, \omega_B, \sigma_B) = sk_B$. Finally, it sends (R_B, ω_B) to $\mathcal{A}$.

Analysis of $G_{4,4}$. Similar to *Analysis of* $G_{1,4}$, under HNF-LWE$_{q,n,\alpha}$ assumption, $G_{4,3}$ and $G_{4,4}$ are computationally indistinguishable. In particular, $\Pr[Q_{4,4}] = \Pr[Q_{4,3}]$ and $\Pr[F_{4,4} \mid \neg Q_{4,4}] = \Pr[F_{4,3} \mid \neg Q_{4,3}] - \mathrm{negl}(n)$.

Similar to *Analysis of* $G_{1,4}$ about $\Pr[Q_{1,4}] = \mathrm{negl}(n)$, under HNF-LWE$_{q,n,\alpha}$ assumption, we have $\Pr[Q_{4,4}] = \mathrm{negl}(n)$.

Similar to *Analysis of* $G_{1,4}$ about $\Pr[F_{1,4} \mid \neg Q_{1,4}] = 1/2 + \mathrm{negl}(n)$, we get $\Pr[F_{4,4} \mid \neg Q_{4,4}] = 1/2 + \mathrm{negl}(n)$.

In summary, $\Pr[F_{4,0}] = \Pr[F_{4,2}] + \mathrm{negl}(n)$ by *Analysis of* $G_{4,1}$ and $G_{4,2}$. By *Analysis of* $G_{4,4}$,

$$\begin{aligned} \Pr[Q_{4,3}] &= \Pr[Q_{4,4}] = \mathrm{negl}(n) \\ \Pr[F_{4,3}] &= \Pr[F_{4,4}] + \mathrm{negl}(n). \end{aligned} \tag{47}$$

Since

$$\begin{aligned} \Pr[F_{4,3}] &= \Pr[F_{4,3} \mid Q_{4,3}] \Pr[Q_{4,3}] \\ &\quad + Pr[F_{4,3} \mid \neg Q_{4,3}] (1 - \Pr[Q_{4,3}]), \end{aligned} \tag{48}$$

thus

$$\Pr\left[F_{4,3}\right] = \Pr\left[F_{4,3} \mid \neg Q_{4,3}\right] - \mathrm{negl}(n). \tag{49}$$

Combining this with *Analysis of* $G_{4,3}$ and $G_{4,4}$, then

$$\Pr\left[F_{4,0}\right] = \Pr\left[F_{4,2}\right] + \mathrm{negl}(n) = \frac{1}{2} + \mathrm{negl}(n). \tag{50}$$

This finishes the proof. □

4.2.3. Type Five Adversary

Theorem 9. *If HNF − LWE$_{q,n,\alpha}$ is hard, the proposed AKE is secure against any PPT Type Five adversary $\mathscr{A}$ under the BR model.*

Proof. We prove the theorem by a series of Games $G_{5,l}$ for $0 \le l \le 4$.

Game $G_{5,0}$. $\mathcal{S}$ randomly selects $A^*, B^* \leftarrow \{1, \ldots, N\}$ and s_{A^*}, $s_{B^*} \leftarrow \{0, \ldots, m\}$, and hopes that the adversary will select $sid^* = (\Pi, R, B^*, A^*, R_{A^*}, (R_{B^*}, \omega_{B^*}))$ as the test session, where R_{A^*} is output by the s_{A^*}th session of party B^* and (R_{B^*}, ω_{B^*}) is output by the s_{B^*}th session of party B^* activated by $Send_1(\Pi, R, B^*, A^*, R_{A^*})$. Then, $\mathcal{S}$ randomly selects $M \leftarrow Z_q^{n\times n}$, honestly generates static public keys for all parties (by randomly choosing $s_A, e_A \leftarrow \mathscr{D}_{\mathbb{Z}^n,\alpha q}$), and simulates security game for $\mathscr{A}$. In particular, $\mathcal{S}$ preserves one table L for random oracle h and responds to queries from $\mathscr{A}$ in the following.

(i) h(in) queries: if there is no tuple (in, out) in L, randomly select an element out $\in \{0,1\}^k$ and add (in, out) to L list. Then send *out* to $\mathscr{A}$.

(ii) $Send_0(\Pi, I, \overline{A}, \overline{B})$: $\mathscr{A}$ initiates a session of $\overline{A}$ with intended partner $\overline{B}$; $\mathcal{S}$ randomly selects $r_A, e_1 \leftarrow \mathscr{D}_{\mathbb{Z}^n,\alpha q}$ and returns $R_A = Mr_A + 2e_1 \bmod q \in Z_q^n$ to $\mathscr{A}$ on behalf of Alice $(\overline{A})$.

(iii) $Send_1(\Pi, R, \overline{B}, \overline{A}, R_A)$: $\mathcal{S}$ randomly selects $r_B, e_2 \leftarrow \mathscr{D}_{\mathbb{Z}^n,\alpha q}$ and computes $R_B = r_B^t M + 2e_2^t \bmod q \in Z_q^n$, k_B, ω_B, and sk_B according to the protocol. At last, send (R_B, ω_B) to $\mathscr{A}$ on behalf of Bob.

(iv) $Send_2(\Pi, I, \overline{A}, \overline{B}, R_A, (R_B, \omega_B))$: $\mathcal{S}$ computes k_A, sk_A by using r_A, s_A according to the protocol.

(v) *SessionKeyReveal(sid)*: let $sid = (\Pi, *, \overline{A}, *, *, *, *)$; $\mathcal{S}$ returns sk_A if session key of *sid* has been generated.

(vi) $Corrupt(\overline{A})$: send Alice's static secret key s_A to $\mathscr{A}$.

(vii) *Test(sid)*: let $sid = (\Pi, I, \overline{A}, \overline{B}, R_A, (R_B, \omega_B))$; if $(\overline{A}, \overline{B}) \neq (A^*, B^*)$, or R_A and R_B are not output by the s_{A^*}th session of A^* and the s_{B^*}th session of B^*, respectively, $\mathcal{S}$ stops. Otherwise, $\mathcal{S}$ randomly selects $b \leftarrow \{0,1\}$, and $sk'_A \leftarrow \{0,1\}^k$. If $b = 0$, $\mathcal{S}$ returns sk'_A; else it returns the real session key sk_A of *sid*.

Analysis of $G_{5,0}$. In this game, $\mathcal{S}$ randomly selects $A^*, B^* \leftarrow \{1, \ldots, N\}$ and $s_{A^*}, s_{B^*} \leftarrow \{1, \ldots, m\}$ independent from $\mathscr{A}$'s view. Thus, the probability that S will not stop in $G_{5,0}$ is at least $1/(m^2 \cdot N^2)$.

Game $G_{5,1}$. $\mathcal{S}$ first computes $\underline{R'_B = r'^t_B M + 2e'_2 \bmod q}$, where $r'_B, e'_2 \leftarrow \mathscr{D}_{\mathbb{Z}^n,\alpha q}$. Then, it acts identically as in $G_{5,0}$ except for the following cases.

(i) $Send_1(\Pi, R, \overline{B}, \overline{A}, R_A)$. If $(\overline{A}, \overline{B}) \neq (A^*, B^*)$, or it is not the s_{B^*}th session of B^*, $\mathcal{S}$ responds to queries as in *Game* $G_{5,0}$. Else, $\mathcal{S}$ randomly selects $d \leftarrow \mathscr{D}_{\mathbb{Z},\alpha q}$ and computes $\underline{R_B = R'_B - d \cdot p_B \bmod q}$, $\underline{k_B = r'^t_B \cdot p_A + r'^t_B \cdot R_A \bmod q}$:

$$\begin{aligned} k_B &= r'^t_B \cdot p_A + r'^t_B \cdot R_A \\ &= r'^t_B \cdot (Ms_A + 2e_A) + r'^t_B \cdot (Mr_A + 2e_1) \\ &= r'^t_B \cdot Ms_A + r'^t_B \cdot Mr_A + r'^t_B \cdot 2e_A + 2r'^t_B \cdot e_1 \\ &\quad \bmod q. \end{aligned} \tag{51}$$

Finally, it honestly computes ω_B, sk_B according to the protocol and sends (y_B, ω_B) to $\mathscr{A}$.

Let $F_{5,l}$ be the event that $\mathscr{A}$ outputs a guess $b' = b$ in *Game* $G_{5,l}$ for $l = 1, 2, 3, 4$.

Analysis of $G_{5,1}$. Similar to *Analysis of* $G_{1,1}$, under HNF-LWE$_{q,n,\alpha}$ assumption, then

$$\Pr\left[F_{5,1}\right] = \Pr\left[F_{5,0}\right] - \mathrm{negl}(n). \tag{52}$$

Game $G_{5,2}$. $\mathcal{S}$ first computes $\underline{R'_A = Mr'_A + 2e'_1 \bmod q}$, where $r'_A, e'_1 \leftarrow \mathscr{D}_{\mathbb{Z}^n,\alpha q}$. Then, $\mathcal{S}$ acts identically as in $G_{5,1}$ apart from the following cases.

(i) $Send_0(\Pi, I, \overline{A}, \overline{B})$. If $(\overline{A}, \overline{B}) \neq (A^*, B^*)$, or it is not the s_{A^*}th session of A^*, $\mathcal{S}$ responds as in *Game* $G_{5,1}$. Else, it randomly selects $c \leftarrow \mathscr{D}_{\mathbb{Z},\alpha q}$, computes $\underline{R_A = R'_A - c \cdot p_A \bmod q}$, and sends R_A to $\mathscr{A}$.

(ii) $Send_2(\Pi, I, \overline{A}, \overline{B}, R_A, (R_B, \omega_B))$. If $(\overline{A}, \overline{B}) \neq (A^*, B^*)$, or it is not the s_{A^*}th session of A^*, $\mathcal{S}$ responds to queries as in *Game* $G_{5,1}$. Otherwise, (R_B, ω_B) is output by the s_{B^*}th session of party B^*; let sk_B be the session key of session $sid = (\Pi, R, \overline{B}, \overline{A}, R_A, (R_B, \omega_B))$; $\mathcal{S}$ sets $\underline{sk_A = sk_B}$. Otherwise, it computes $\underline{k_A = (p_B + R_B) \cdot r'_A \bmod q}$. At last, $\mathcal{S}$ computes sk_A according to the protocol.

Analysis of $G_{5,2}$. Similar to *Analysis of* $G_{1,2}$, under HNF-LWE$_{q,n,\alpha}$ assumption, we have

$$\Pr\left[F_{5,2}\right] = \Pr\left[F_{5,1}\right] - \mathrm{negl}(n). \tag{53}$$

Game $G_{5,3}$. $\mathcal{S}$ first randomly chooses $\underline{R'_A \leftarrow Z_q^n}$. Then it acts identically as in $G_{5,2}$ with the following exceptions.

Table 1: Comparison between previous AKEs and our proposed AKE.

AKE	Comp.	Sec.	Model	Rom	Assum.	Qua.
NAXOS [7]	$4E$	∘	eCK, wPFS	Rom	GDH	No
CMQV [34]	$3E$	∘	eCK, wPFS	Rom	GDH	No
HMQV [5]	$3E$	∘	CK^+, wPFS	Rom	GDH	No
FSX [26]	0	OW-CCA	CK^+	Rom	RLWE	Yes
FSXY [27]	0	CCA^+	CK^+	×	(R-)LWE	Yes
BCN [18]	0	CPA	ACCE	Rom	RLWE	Yes
Ours	0	∘	BR, wPFS	Rom	LWE	Yes

(i) $Send_2(\Pi, I, \overline{A}, \overline{B}, R_A, (R_B, \omega_B))$. If $(\overline{A}, \overline{B}) \neq (A^*, B^*)$, or it is not the s_{A^*} th session of A^*, or (R_B, ω_B) is output by the s_{B^*} th session of party B^*, $\mathcal{S}$ responds to query as in *Game* $G_{5,2}$. Else, it randomly selects $\underline{sk_A \leftarrow \{0,1\}^k}$ as the session key.

Analysis of $G_{5,3}$. Similar to Analysis of $G_{2,4}$, $G_{2,5}$, and $G_{2,6}$, under $\mathrm{LWE}_{q,n,\alpha}$ assumption, we have

$$\Pr\left[F_{5,3}\right] = \Pr\left[F_{5,2}\right] - \mathrm{negl}\,(n). \tag{54}$$

Game $G_{5,4}$. $\mathcal{S}$ randomly selects $\underline{R'_B \leftarrow Z_q^n}$ and acts identically in $G_{5,3}$ with the following exceptions.

(i) $Send_1(\Pi, R, \overline{B}, \overline{A}, R_A)$. If $(\overline{A}, \overline{B}) \neq (A^*, B^*)$ or it is not the s_B^*th session of B^*, $\mathcal{S}$ responds to the query as in *Game* $G_{5,3}$. Otherwise, $\mathcal{S}$ randomly chooses $d \leftarrow \mathcal{D}_{\mathbb{Z},\alpha,q}$ and computes $\underline{R_B = R'_B - d \cdot p_B \bmod q}$. Then $\mathcal{S}$ randomly selects $\underline{k_B \leftarrow Z_q}$ and computes $\underline{\omega_B, \sigma_B}$ as described in protocol. If A has made a h query $h(\overline{A}, \overline{B}, R_A, R_B, \omega_B, \sigma_B)$, $\mathcal{S}$ stops. Otherwise, it randomly selects $\underline{sk_B \leftarrow \{0,1\}^k}$ and sets $h(\overline{A}, \overline{B}, R_A, R_B, \omega_B, \sigma_B) = sk_B$. At last, it $\underline{\text{returns } (R_B, \omega_B)}$ to $\mathcal{A}$.

Analysis of $G_{5,4}$. Similar to *Analysis of* $G_{1,4}$, under $\text{HNF-LWE}_{q,n,\alpha}$ assumption, we have $\Pr[F_{5,4}] = \Pr[F_{5,3}] - \mathrm{negl}(n)$.

Similar to *Analysis of* $G_{1,4}$ about $\Pr[F_{1,4}] = 1/2 + \mathrm{negl}(n)$, under $\text{HNF-LWE}_{q,n,\alpha}$ assumption, we have

$$\Pr\left[F_{5,4} \mid \neg Q_{5,4}\right] = \frac{1}{2} + \mathrm{negl}\,(n). \tag{55}$$

In summary, $\Pr[F_{5,0}] = \Pr[F_{5,2}] + \mathrm{negl}(n)$ by *Analysis of* $G_{5,1}$ and $G_{5,2}$. By *Analysis of* $G_{5,4}$, $\Pr[F_{5,3}] = \Pr[F_{5,4}] + \mathrm{negl}(n)$. Combining this with *Analysis of* $G_{5,3}$ and $G_{5,4}$, we have

$$\Pr\left[F_{5,0}\right] = \Pr\left[F_{5,2}\right] + \mathrm{negl}\,(n) = \frac{1}{2} + \mathrm{negl}\,(n). \tag{56}$$

This completes the proof. □

5. Comparison of Performance and Security

At present, there is a handful of results on lattice-based AKE under the BR model. Table 1 compares our protocol with other AKEs in terms of computational complexity and security. For simplicity, Comp. means computation methods. "E" denotes the exponentiation. "0" means matrix-vector multiplication, not exponentiation operation. Sec. stands for security level. For example, CCA^+ means CCA security with high min-entropy keys [27]. Denote by ∘ no security level. Model means security model. For example, CK^+ [5] denotes modified CK security model [4]. ACCE means authenticated and confidential channel establishment [18]. Rom means random oracle model. × denotes no Rom. Assum. denotes underlying hardness assumptions. GDH stands for gap Diffie-Hellman assumptions. Qua. denotes quantum attack. Yes means resisting quantum attack; No means suffering quantum attack.

Note that NAXOS [7], CMQV [34], and HMQV [5] referred to exponentiation computation and achieve AKE security with wPFS in GDH assumption which indicated that they were vulnerable to quantum attack although they are secure in stronger model. Compared with NAXOS [7], CMQV [34], and HMQV [5] based on GDH assumption, our protocol has much more advantages in terms of computation because our protocol grasps matrix-vector multiplication besides resistance to quantum attack. In terms of Sec., [18, 26, 27] captured OW-CCA, CCA^+, and CPA security without wPFS. But ours achieves security and wPFS without authentication tools under the BR model.

The new protocol has a good balance between computation and security.

6. Conclusion

This paper first proposes an AKE from the LWE problem. The AKE is simple since it does not involve any other cryptographic primitives (e.g., MAC, signature) to achieve authentication and depends on solely the LWE problem in the worst-case (e.g., SVP and SIVP [12–14]). Security analysis with wPFS is proved to resist five kinds of adversaries under the BR model and it might be appealing in specific applications.

This paper also motivates interesting open problems, such as an attack on it, converting it to one AKE under the CK model. If our lattice-based AKE is improved, it may achieve CPA and CCA security with wPFS, PFS, KCI, and so on under the CK, eCK, and CK^+ model. Maybe there exists a SFA on our protocol. We do not study them here and leave them as the future works.

Appendix

BR Model for AKE

This section outlines the BR model, for further details the reader is referred to [1].

Sessions. Assume that n is security parameter and $N = N(n) \in \mathbb{Z}$ denotes the maximum number of honest parties, each of whom is uniquely identified by integers in $[N] = \{1, \ldots, N\}$, and has a pair of static public/secret keys (p_i, s_i). An execution of an AKE was called a *session sid* $= (\Pi, I, i, j)$ or $sid = (\Pi, I, i, j, X_i, Y_j)$ [34] and its *matching session* is $\widetilde{sid} = (\Pi, R, j, i, X_i, Y_j)$ [34]. Π stands for an AKE protocol.

Adversarial Powers. An adversary $\mathcal{A}$ is defined to be a probabilistic polynomial time (PPT) Turing machine that controls all communications between parties including obtaining static secret keys via oracle queries below. To capture wPFS [5], $\mathcal{A}$ can corrupt an honest party of a session sid^*. In accordance with our security model, *Send* query [1, 4, 35] included $Send_0$, $Send_1$, and $Send_2$.

(i) $Send_0(\Pi, I, i, j)$ makes $\mathcal{A}$ perform the first step of our protocol and create a session with i as an initiator. The oracle returns a message X_i to party j.

(ii) $Send_1(\Pi, R, \mathrm{j}, i, X_i)$ makes $\mathcal{A}$ perform the second step of our protocol and use message X_i to create a session with party j as a responder. The oracle returns a message Y_j to party i.

(iii) $Send_2(\Pi, R, i, j, X_i, Y_j)$ makes $\mathcal{A}$ perform the third step of our protocol and send party i message Y_j to complete a session intended for $Send_0(\Pi, I, i, j)$ query.

(iv) $SessionKeyReveal(sid)$: the adversary obtains the session key for a session sid if sid holds a session key.

(v) $Corrupt(i)$: the oracle returns the static secret key of the honest party i to $\mathcal{A}$.

(vi) $Test(sid^*)$: the oracle chooses a bit $b \leftarrow \{0, 1\}$; if $b = 0$, $\mathcal{A}$ receives a uniformly chosen random value and otherwise the actual session key.

The adversary $\mathcal{A}$ can only once query the test session which was *freshness* [35].

AKE Security. The security of a two-pass AKE protocol Π is defined via a series of games in which $\mathcal{A}$ makes any sequence of queries to the oracles above. The game ends when $\mathcal{A}$ outputs a guess b' of b. The advantage of $\mathcal{A}$ in Π is defined as $\mathrm{Adv}^{\mathrm{AKE}}_{\Pi,\mathcal{A}} = \Pr[b' = b] - 1/2$ in attacking Π which declares that $\mathcal{A}$ wins the game if $b' = b$. Namely, AKE Π is *AKE secure* if, for any PPT adversary $\mathcal{A}$, $\mathrm{Adv}^{\mathrm{AKE}}_{\Pi,\mathcal{A}}$ is *negligible*.

For convenience, i stands for Alice and j stands for Bob through the paper.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China (NSFC) (no. 61370194).

References

[1] M. Bellare and P. Rogaway, "Entity authentication and key distribution," in *Advances in Cryptology—CRYPTO '93*, vol. 773 of *Lecture Notes in Computer Science*, pp. 232–249, Springer, Berlin, Germany, 1994.

[2] M. Bellare and P. Rogaway, "Provably secure session key distribution: the three party case," in *Proceedings of the 27th Annual ACM Symposium on Theory of Computing (STOC '95)*, pp. 57–66, ACM Press, Las Vegas, Nev, USA, May-June 1995.

[3] M. Bellare, D. Pointcheval, and P. Rogaway, "Authenticated key exchange secure against dictionary attacks," in *Advances in Cryptology—EUROCRYPT 2000*, vol. 1807 of *Lecture Notes in Computer Science*, pp. 139–155, Springer, Berlin, Germany, 2000.

[4] R. Canetti and H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels," in *Advances in Cryptology—EUROCRYPT 2001*, pp. 453–474, Springer, Berlin, Germany, 2001.

[5] H. Krawczyk, "HMQV: a high-performance secure Diffie-Hellman protocol (extended abstract)," in *Advances in cryptology—CRYPTO 2005*, vol. 3621 of *Lecture Notes in Computer Science*, pp. 546–566, Springer, Berlin, Germany, 2005.

[6] C. Cremers, "Examining indistinguishability-based security models for key exchange protocols: the case of CK, CK-HMQV, and eCK," in *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*, pp. 80–91, ACM, Hong Kong, March 2011.

[7] B. LaMacchia, K. Lauter, and A. Mityagin, "Stronger security of authenticated key exchange," in *Provable Security*, vol. 4784 of *Lecture Notes in Computer Science*, pp. 1–16, Springer, Berlin, Germany, 2007.

[8] A. P. Sarr, P. Elbaz-Vincent, and J. C. Bajard, "A new security model for authenticated key agreement," in *Security and Cryptography for Networks*, pp. 219–234, Springer, Berlin, Germany, 2010.

[9] M. O. Rabin, *Digitalized Signatures and Public-Key Functions as Intractable as Factorization*, Massachusetts Institute of Technology, Cambridge Laboratory for Computer Science, 1979.

[10] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol. 31, no. 4, pp. 469–472, 1985.

[11] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Journal on Computing*, vol. 26, no. 5, pp. 1484–1509, 1997.

[12] C. Gentry, C. Peikert, and V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," in *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC '08)*, pp. 197–206, Victoria, Canada, May 2008.

[13] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," in *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC '05)*, pp. 84–93, Baltimore, Md, USA, May 2005.

[14] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," *Journal of the ACM*, vol. 56, no. 6, article 34, 2009.

[15] C. Peikert, "Public-key cryptosystems from the worst-case shortest vector problem," in *Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC '09)*, pp. 333–342, ACM, Bethesda, Md, USA, 2009.

[16] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," in *Advances in Cryptology—EUROCRYPT 2010: 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30–June 3, 2010. Proceedings*, H. Gilbert, Ed., vol. 6110 of *Lecture Notes in Computer Science*, pp. 1–23, Springer, Berlin, Germany, 2010.

[17] C. Peikert, "Lattice cryptography for the internet," in *Post-Quantum Cryptography*, vol. 8772 of *Lecture Notes in Computer Science*, pp. 197–219, Springer International Publishing, Cham, Switzerland, 2014.

[18] J. W. Bos, C. Costello, M. Naehrig, and D. Stebila, "Post-quantum key exchange for the TLS protocol from the ring learning with errors problem," in *Proceedings of the 36th IEEE Symposium on Security and Privacy (SP '15)*, pp. 553–570, San Jose, Calif, USA, May 2015.

[19] J. Katz and V. Vaikuntanathan, "Smooth projective hashing and password-based authenticated key exchange from lattices," in *Advances in Cryptology—ASIACRYPT 2009*, M. Matsui, Ed., vol. 5912 of *Lecture Notes in Computer Science*, pp. 636–652, Springer, Berlin, Germany, 2009.

[20] J. Ding and X. Lin, "A simple provably secure key exchange scheme based on the learning with errors problem," IACR Cryptology ePrint Archive 2012/688, 2012.

[21] X. Lei and X. Liao, "NTRU-KE: a lattice-based public key exchange protocol," *IACR Cryptology ePrint Archive*, vol. 2013, article 718, 2013.

[22] S. Wang, Y. Zhu, D. Ma, and R. Feng, "Lattice-based key exchange on small integer solution problem," *Science China. Information Sciences*, vol. 57, no. 11, pp. 1–12, 2014.

[23] L. Wulu, "A key exchange scheme based on lattice," in *Proceedings of the 11th IEEE International Conference on Dependable, Autonomic and Secure Computing (DASC '13)*, pp. 100–106, Chengdu, China, December 2013.

[24] J. Zhang, Z. Zhang, J. Ding et al., "Authenticated key exchange from ideal lattices," in *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 719–751, Springer, Sofia, Bulgaria, April 2015.

[25] V. Singh, "A practical key exchange for the internet using lattice cryptography," IACR Cryptology ePrint Archive 2015/138, 2015.

[26] A. Fujioka, K. Suzuki, K. Xagawa, and K. Yoneyama, "Practical and post-quantum authenticated key exchange from one-way secure key encapsulation mechanism," in *Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security (ASIA CCS '13)*, pp. 83–94, ACM, May 2013.

[27] A. Fujioka, K. Suzuki, K. Xagawa, and K. Yoneyama, "Strongly secure authenticated key exchange from factoring, codes, and lattices," *Designs, Codes and Cryptography*, vol. 76, no. 3, pp. 469–504, 2015.

[28] J. Ding, S. Alsayigh, R. V. Saraswathy et al., "Leakage of signal function with reused keys in RLWE key exchange," Cryptology ePrint Archive Report 2016/1176, 2016.

[29] B. Gong and Y. Zhao, "Small field attack, and revisiting RLWE-based authenticated key exchange from Eurocrypt'15," Cryptology ePrint Archive Report 2016/913, 2016, http://eprint.iacr.org/2016/913.

[30] R. Lidl and H. Niederreiter, *Instruction to Algebra and Finite Fields*, Cambridge University Press, 2000, http://www.cambridge.org.

[31] Z. Brakerski, A. Langlois, C. Peikert, O. Regev, and D. Stehlé, "Classical hardness of learning with errors," in *Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC '13)*, pp. 575–584, ACM, June 2013.

[32] B. Applebaum, D. Cash, C. Peikert et al., "Fast cryptographic primitives and circular-secure encryption based on hard learning problems," in *Advances in Cryptology—CRYPTO 2009*, pp. 595–618, Springer, Berlin, Germany, 2009.

[33] D. Micciancio and O. Regev, "Worst-case to average-case reductions based on Gaussian measures," *SIAM Journal on Computing*, vol. 37, no. 1, pp. 267–302, 2007.

[34] B. Ustaoglu, "Obtaining a secure and efficient key agreement protocol from (H)MQV and NAXOS," *Designs, Codes, and Cryptography*, vol. 46, no. 3, pp. 329–342, 2008.

[35] K. K. R. Choo, C. Boyd, and Y. Hitchcock, "Examining indistinguishability-based proof models for key establishment protocols," in *Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security*, pp. 585–604, Springer, Chennai, India, December 2005.